A COMPUTER-BASED MANAGEMENT INFORMATION SYSTEM FOR A HUMANISTIC UNDERGRADUATE ELEMENTARY TEACHER EDUCATION PROGRAM

By

BERNICE DOERR

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL OF
THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1975

For my parents

ACKNOWLEDGMENTS

I wish to express my deep gratitude to the many people who devoted time and effort to this study. Dr. William D. Hedges, my chairman, was a constant source of assistance throughout the study. Dr. Frank D. Vickers provided support, direction, and many helpful resources. Dr. Vynce Hines and Dr. William Ware, who both provided support and assistance, were especially helpful editorially.

I also wish to thank the faculty and students on Team III for their support, cooperation, and assistance in this study. Without their help, it would not have been possible.

The assistance and support of my fellow-students is also greatly appreciated.

I am deeply indebted to the late Dr. Charles M. Bridges, Jr. who encouraged me to pursue doctoral studies at the University of Florida and served as my chairman until his death. His guidance, counseling, and constant support were a prime reason for my successful completion of graduate studies.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS. iii

LIST OF TABLES vi

ABSTRACT vii

CHAPTER
I. INTRODUCTION 1
The Problem. 2
Assumptions and Limitations. 2
Need for the Study 3
Relevance of the Study 4
Definition of Terms. 4
Organization of the Study. 7

II. REVIEW OF RELATED LITERATURE AND DESCRIPTION OF THE CURRENT SYSTEM. 8
Review of Related Literature 8
Introduction 8
Definitions. 9
Development of Information Systems in Education. 10
Management Information Systems 11
Humanistic Field-based Teacher Education Programs. 20
Description of the Current Management System 24

III. PROCEDURE. 28
Overview 28
Specify the Goals and Objectives of the System 28
Develop the System 30
Test the System. 32
Revise the System. 35
Evaluate the System. 36

IV. DESCRIPTION AND EVALUATION OF THE COMPUTER-BASED MANAGEMENT INFORMATION SYSTEM AS TESTED 38
Description. 38
Evaluation 40

V. SUMMARY, CONCLUSIONS AND RECOMMENDATION FOR FUTURE RESEARCH AND DEVELOPMENT. 50
Summary. 50
Conclusions. 51
Recommendations for Future Research and Development. 53

TABLE OF CONTENTS - Continued

Page

APPENDICES. 56

APPENDIX A CEP MANAGEMENT INFORMATION SYSTEM EVALUATION . . 56

APPENDIX B MEMO TO FACULTY MEMBERS. 60

APPENDIX C PROGRAMS IN THE COMPUTER-BASED MANAGEMENT INFORMATION SYSTEM 62

APPENDIX D CEP MANAGEMENT INFORMATION SYSTEM USERS' MANUAL. 133

BIBLIOGRAPHY. 242

BIOGRAPHICAL SKETCH 246

LIST OF TABLES

Page

TABLE 1 CEP MANAGEMENT INFORMATION SYSTEM EVALUATION RESPONSES TO ITEM 1. 43

TABLE 2 CEP MANAGEMENT INFORMATION SYSTEM EVALUATION RESPONSES TO ITEM 2. 45

TABLE 3 CEP MANAGEMENT INFORMATION SYSTEM EVALUATION RESPONSES TO ITEM 3. 46

TABLE 4 CEP MANAGEMENT INFORMATION SYSTEM EVALUATION RESPONSES TO ITEM 4. 47

TABLE 5 CEP MANAGEMENT INFORMATION SYSTEM EVALUATION RESPONSES TO ITEM 5. 48

Abstract of Dissertation Presented to the Graduate Council
of the University of Florida in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

A COMPUTER-BASED MANAGEMENT INFORMATION SYSTEM FOR A HUMANISTIC UNDERGRADUATE ELEMENTARY TEACHER EDUCATION PROGRAM

By

Bernice Doerr

August, 1975

Chairman: William D. Hedges
Major Department: Foundations of Education

In recent years numerous innovative teacher training programs which incorporate new knowledge and materials have been developed. One of these programs is the humanistic field-based teacher education program at the University of Florida called the Childhood Education Program (CEP).

Underlying assumptions of CEP require that traditional courses be replaced by seminars, field experience, and competency in substantive areas. Students work at their own pace as they complete their individually tailored programs. Detailed records, used in the continuous evaluation of their progress, are maintained by the faculty.

This study examines the feasibility of developing and operating a computer-based management information system for CEP to provide readily accessible and reliable information about the team and students in it.

Development of the system was conducted in terms of the following: specifying the goals and objectives of the system, developing the system, testing the system, and evaluating the system.

Initially, a basic understanding of the program was obtained by the investigator in order to enable specification of the goals and objectives. Then the actual development of the system was begun. The development procedure was broken down into the following activities: specifying the student data base, writing the software using PL/1, designing procedures for the collection and dissemination of information, and writing a users' manual.

A test of the system was conducted for a period of one quarter on a seminar of twenty-eight students and the faculty members on their team. The computer-based system was operated parallel with the current management system. Although no major changes were made in initial programs in the system, new programs were added when their need became apparent.

Faculty evaluation of the system was conducted at the end of the quarter using the CEP Management Information System Evaluation form. The result: all faculty members would like to see the system in operation for the entire team since it provided timely and accurate information about students and the team. This information could be used for counseling, decision making, and research.

Although faculty members indicated they wanted to see the system in operation for the entire team, they were reluctant to use it despite prodding from the investigator. This reluctance probably stems from a general fear about the use of the computer and the limitations imposed by the test. Hopefully this reluctance can be eliminated by further education during full scale implementation.

CHAPTER I

INTRODUCTION

During the last decade it became apparent that teacher education programs were not meeting the needs of our rapidly changing technological society. Research and development activities generated new knowledge and materials, but these were not adequately incorporated into teacher training programs. It became increasingly apparent that there was a need to incorporate these new materials into teacher training models. Consequently, the U.S. Office of Education funded a number of training models. Other innovative programs, not funded, were devised as well. One of these programs was the humanistic field-based teacher education program at the University of Florida called the Childhood Education Program (CEP). CEP is a humanistic program since it was "designed to help each student find his own best ways of teaching" (Wass et al., Preface). This study concerns CEP.

The underlying assumptions of this program require that traditional courses be replaced by seminars, field experience, and competency in substantive (subject) areas. Within these procedures, students work at their own pace as they complete the various learning activities in their individually tailored programs. In the process, seminar leaders are required to maintain detailed records of each student's progress. These records are a vital part of the continuous evaluation procedures which must characterize such a program if it is to be maximally effective.

The Problem

This study examines the feasibility of developing and operating a computer-based management information system which maintains the underlying characteristics of CEP. Specifically this study addresses the following questions:

1. What is the flow of information in the current management system? Should this flow be altered? If so, how and why?
2. What types of data are currently being collected and why are they needed? If there is redundancy and inefficiency in the data currently being collected, how can this be corrected? What other types of data are needed? Why are they needed?
3. When are these data needed? In what forms are they needed? Who needs these data? How are and how should they be used?

Assumptions and Limitations

This study examines the feasibility of establishing and operating a computer-based management information system for CEP. An initial assumption was that there is sufficient similarity between and among different teams in the program to allow one team to serve as a model for the development of the system. Thus, minor modifications to the system should permit it to be used with other teams.

Ideally the system developed would have been tested on an entire team of one hundred and twenty students. However, limited resources and the assignment of faculty across teams made this impractical. Thus, the trial run of the system was conducted for one quarter on one seminar group of twenty-eight students and faculty members on this team.

The computer-based system was operated parallel with the then current management system. Although this type of test provided checkup and backup capabilities, it was in some respects an artificial situation. Because the current management system was not to be changed it imposed an inefficient flow of information about learning activities, field experiences, and community experiences into the computer-based system. In some cases, this inefficiency was caused by unnecessary human intervention at various stages before the data entered the system. Thus, the full capabilities of the computer-based system could not be realized.

All of the problems associated with the operation of the system on a year-round basis probably were not encountered during the limited test period. However, the testing has been sufficiently comprehensive to enable the investigator to document major strengths and limitations of the computer-based management system as designed and implemented.

Need for the Study

At every level of education from the elementary school to the university, records of students' progress must be maintained. The quantity and types of data kept on each student vary with the level and type of program. Perhaps, the most demanding type of record keeping is in a program where students progress at their own pace in their own individual manner. One such program in teacher education is CEP.

Currently, seminar leaders and substantive panel members are plagued by an antiquated system of record keeping including those forms used in the final, midpoint, and quarterly contract evaluations. Some forms may be unnecessary. Other forms may contain data which is superfluous to the program. There is duplication of effort in maintaining various types of data. Valuable time which could be spent interacting with the students is wasted.

Some of the current record-keeping problems could be alleviated by the development of a computer-based management information system. Basic student data would need to be entered into the system only one time. Once stored in the system, these data plus periodic status data could be used to generate student progress reports and special summary reports as needed. These data could be used for both formative and summative evaluation of CEP. Current record-keeping procedures make such evaluation of the program a tedious, if not impossible, task.

Relevance of the Study

This study documents the feasibility of establishing and operating a computer-based management information system for humanistic teacher education programs similar to this one. Perhaps, more importantly, it indicates problems and possible solutions which may arise when an attempt is made to join two important forces in education today, i.e., technology and humanism. Technology via the computer and humanism are considered by many educators to be diametrically opposed to one another. What, in fact, does happen to basic components of a humanistic program with the advent of a computer-based management information system?

Definition of Terms

CEP - An acronym for the Childhood Education Program.

Program - A term used for addressing the Childhood Education Program.

MIS - An acronym for management information system.

Seminar - A group made up of approximately thirty students and one faculty member. It provides counseling, guidance, discussion about educational theories and practices, and training in communication skills.

Seminar leader - A faculty member who serves as the personal counselor, academic advisor, and teacher for each student in his seminar for their entire stay in the program.

Substantive panel - A group of instructors who represent the various areas of teaching methods such as art, language arts, mathematics, music, science, social studies, reading, and specialists in curriculum, psychological, and social foundations.

Team - A group composed of the team leader, four seminar leaders, the substantive panel members, and approximately one hundred and twenty students.

Team meeting - A biweekly meeting, which is attended by faculty members and student representatives from each seminar, where decisions concerning policies of the team are made.

Learning activity - A number of activities of various forms must be completed in each of the substantive areas. Some activities are required, others are chosen from a list of alternatives, and still others are negotiated.

Beyond the minimum - A program requirement for which students do more than the minimum number of learning activities in three substantive areas.

Field experience - A direct school experience where students are continuously involved with increasing time commitments as they progress through the program. Through the field experience students become aware of what they need to know and cognizant of those skills they must develop in order to become a teacher.

Community experience - An alternate form of field experience where students are placed for one quarter with some community agency to become more aware of the community in which they are working.

Community session - A regularly scheduled meeting of all seminars within a team where a wide variety of programs are presented.

Quarterly contract - A "quarter contract" is submitted by each student to his seminar leader outlining his expected progress for the quarter. The student's actual progress is reviewed with the seminar leader at the end of the quarter.

Midpoint evaluation - A conference at the midpoint of a student's program involving the student, his seminar leader, and one or two faculty members from the substantive panel. These individuals examine the student's progress, discuss her (his) field experiences, and plans for the remainder of the program.

Final evaluation - A conference, which is similar to the midpoint evaluation, during which the student's entire progress through the program is reviewed. At this time it is determined if the student is ready to teach.

Intensive - A level of field experience, usually occurring during the final quarter in the program, where students spend five weeks of full-time teaching.

DD statement - An acronym for data definition statement. By altering the DD statement different input/output devices or data sets may be designated for a file.

Structure - A hierarchical collection of data that possess a logical relationship to one another. The structure PERSONAL contains the entire collection of data stored on each student.

Organization of the Study

Chapter I consists of the introduction, statement of the problem, assumptions and limitations, need for the study, relevance of the study, and definition of terms. In Chapter II a comprehensive review of the relevant literature on both management information systems used in education and humanistic teacher education and description of the current management system are set forth. Chapter III describes the procedures used in the development of the system. In Chapter IV a description and evaluation of the computer-based management information system which was tested are presented. Chapter V presents the summary, conclusions, and recommendations for future research and development.

CHAPTER II

REVIEW OF RELATED LITERATURE AND DESCRIPTION OF THE CURRENT SYSTEM

Review of Related Literature

Introduction

Initially it is necessary to examine the current study in light of changes in education and society as a whole. According to Molnar (1972),

> our Society is said to be in a transition from being one based upon industrial production to one based upon knowledge. This new emphasis on knowledge, in turn, places greater demands on education. More and more, education is a prerequisite to participation in the knowledge society. Many schools and universities, are reevaluating their curricula, [and] examining the needs of their students. (p. 3)

As new curricula developed, "information problems -- ranging from problems of collecting, storing, communicating, retrieving, and displaying information to problems of receiving, learning, and using information" (Goodlad et al., 1966, p. 10) arose. In view of this situation, Goodlad and others reported:

> The continued use of antiquated information processing procedures in the face of the growing size and complexity of American education constitutes a gross anachronism. Education is in large measure an information processing system. The right data at the right moment give teachers and administrators the edge they need for making calculated rather than uninformed decisions. Wisdom and information are a powerful team. Computers can provide the information; educators must provide the wisdom. (p. 20)

Thus, the stage was set for the rapid adoption of the computer for administration, instruction, and research in education. By 1972 nearly all universities and more than a third of the four-year colleges in the United States provided computing services (Molnar, 1972, p. 6).

Definitions

What is a currently acceptable definition of a management information system (MIS)? According to Kennevan (1970),

> A management information system is an organized method of providing past, present and projection information relating to internal operations and external intelligence. It supports the planning, control and operational functions of an organization by furnishing uniform information in the proper timeframe to assist the decision-making process. (p. 63)

From a more detailed perspective, Teichroew and Sayani (1971) stated that

> an "information system" consists of two subsystems: a management system and an information processing system. The management system consists of the organization, its objectives, the individuals or groups in it and the rules and procedures under which they work. The information processing system is the subsystem which consists of hardware, programs, non-computerized procedures, etc., that accomplish the storage, processing and communication of information necessary for the functioning of the management system. An essential element in this view is that the information processing system (IPS) must, or at least should, be designed to serve the management system. (p. 25)

Thus, a management information system can be differentiated from a general "information system" by the fact that it "organizes data into meaningful reports that are designed to fulfill defined information needs for management" (Mathews et al., 1972, p. 4).

Although the words "information" and "data" are often used interchangeably, a distinction can be made between them. Data are unrelated and uninterpreted facts, whereas information is obtained by organizing and analyzing data into a useful form which increases knowledge (Levin, 1956).

Development of Information Systems in Education

Van Dusseldorp (1971) noted that the development of computerized information systems in education has moved through four observable approaches since the mid 1950s. First was the applications approach. Separate applications usually of a clerical nature, such as payroll, school scheduling, and grade reporting, were computerized. Second came the integrated systems approach. In order to integrate information from various applications, many data elements had to be redefined and coded according to another system. Third was the total systems approach. Initially, all information needed by the educational agency was defined. Duplication was avoided since data were gathered, stored, and processed through a central system. The fourth approach was the management information system. Here, the primary emphasis is to develop systems which provide the information needed by management for control, decision making, and planning. In addition, these systems

> provide educational research with a vehicle for better identifying the more significant and critical areas requiring the attention of the educational research community as well as a considerably expanded data base upon which to draw for their research efforts. (Lawrence et al., 1971, p. 296)

Management information systems must be of a dynamic nature since they operate in the real world. According to Hayman (1974), this environment has certain effects on the system.

> 1) It turns out in practice that initial MIS development never results in outputs which fully meet decision needs, so that long-term developmental activity (an evolutionary approach) is essential; 2) organizational needs and functions change over time, and the MIS must change with them; 3) technology changes, and the system must be updated. (p. 63)

Management Information Systems

Management information systems have been developed at all levels of education from the elementary school classroom to the university. These systems can be classified into two general categories: "limited purpose" management information systems whose entire scope of operation is confined to student records and "general purpose" management information systems whose scope of operation usually includes student data, personnel data, facilities data, program data, and financial data. User needs dictate the type of system which is developed.

Development of "limited purpose" management information systems often appears to be related to individualized instruction. Hansen and Harvey (1970) describe a pattern of computer development in the public schools which closely parallels this development process. An initial commitment to individualized instruction is followed by the development of individually prescribed instruction (IPI). Then management systems are developed to keep track of student progress.

The literature is thoroughly documented with descriptions of "limited purpose" systems. At the elementary and secondary school levels, these systems have not often been developed on their own. Rather, they are usually subsumed under the operation of computer managed instruction (CMI). CMI is more inclusive since in addition to reporting, it includes test scoring, diagnosis, and prescription (Baker, 1971).

Individually Prescribed Instruction (IPI), which was developed at the Learning Research and Development Center (LRDC) at the University of Pittsburgh by Cooley and Glaser, adapts instructional practices to individual requirements. According to the model for IPI,

> 1. The goals of learning are specified in terms of observable student behavior and the conditions under which this behavior is to be exercised.
> 2. Diagnosis is made of the initial capabilities with which the learner begins a particular course of instruction. The capabilities that are assessed are those relevant to the forthcoming instruction.
> 3. Educational alternatives adaptive to the initial profile of the student are presented to him. The student selects or is assigned one of these alternatives.
> 4. Student performance is monitored and continuously assessed as the student proceeds to learn.
> 5. Instruction proceeds as a function of the relationship between measures of student performance, available instructional alternatives, and criteria of competence.
> 6. As instruction proceeds, data are generated for monitoring and improving the instructional system. (1968, pp. 2-3)

Initial operation of IPI was nonautomated. A management and information system (IPI/MIS), whose design was based on the needs of the staff, was later developed. The four major functions of this system are to "(1) collect data; (2) monitor student progress; (3) provide prescription information; (4) diagnose student difficulty" (Cooley and Glaser, p. 17).

Another system very similar to IPI/MIS is the Instructional Management System (IMS) which was developed jointly by the Systems Development Corporation and the Southwest Regional Laboratory for Educational Research and Development. This system was designed to "help the teacher monitor the progress of her pupils and make decisions on the pace of instruction, the grouping of children, the sequence of lessons and the individualization of instruction" (Geddes and Kooi, 1969, p. 337). Students are tested once or twice a week; results are processed by an optical scanner; and computer generated reports are given to the teacher. Other reports, such as weekly summaries, are also provided (Geddes and Kooi, 1969; Silberman, 1969).

Project PLAN, Program for Learning in Accordance with Needs, was developed jointly by the American Institute for Research and Westinghouse Learning Corporation. In the PLAN system, described by Flanagan (1970),

> the daily processing and weekly status reports constitute the tactics of the interaction of the student and teacher with the computer. . . .
>
> The major strategy of the PLAN educational system is related to the guidance and individual planning functions, the program of studies, and the analysis of students' results on the module and achievement tests. These aspects are also heavily dependent on the computer, primarily for doing scoring, recordkeeping, matching, and making predictions and indicating probabilities. The computer thus becomes a very valuable resource for the teacher and student in planning and guiding the student's educational program. (p. 15)

A system, more limited than those previously described, called COMPUTRAC was developed at Florida State University to help teachers in monitoring student classroom performance. Daily input into the system is used to produce a weekly cumulative summary of each student's progress. This helps the teacher determine students needing attention. The six-week class progress file and bi-yearly analysis of performance on objectives are used to identify ineffective learning routines (Healy, 1971).

The preceding projects demonstrated the use of "limited purpose" management information systems in public schools. Attention is now directed toward literature which documents the use of this type of system in higher education.

The School of Engineering at the University of Pittsburgh developed a computerized information system to provide answers to questions about students in the school. The data base for this system is quite broad. It is used to generate: academic status statistics, student profile inquiries, student registration advisory reports, enrollment statistics, physical plant utilization reports, and statistical analyses of various items in the data base (Holzman and Turksen, 1971).

Adoption of a multidimensional approach to student evaluation by the University of Tennessee College of Medicine necessitated the development of a computerized student evaluation system to generate current and accurate reports about student progress. Stefanu and others (1975) gave a brief description of the system.

> A comprehensive current record of every student's performance is readily available to those faculty members responsible for guiding the prospective physician or making decisions concerning their program advancement. The multiphasic evaluations recorded by each course director and compiled as part of the report-generating model facilitate the identification of problem areas as well as strong points exhibited by the students in their work at the Medical Units. Also, immediate feedback is given to the students concerning the evaluations which they receive from their instructors. The prospective physicians then have the means to determine their own progress and make adjustments necessary to improve both their academic and professional status. (p. 203)

As mentioned earlier, "general purpose" management information systems have also been developed. Although these systems go beyond the scope of the current study, they often do include features which can be incorporated into "limited purpose" systems. The literature which is reported on "general purpose" management information systems documents the fact that there is a national concern for the development of such systems to aid decision making.

PMIS, a Planning and Management Information System, was a project of the Council of Great City Schools. The complete self-contained system which was developed "has its own data structures, data loading, editing and modification software, its own query languages, and detailed specifications for use and application" (Fiddleman and Gorman, 1972, p. 89). The design of the system was described by Fiddleman and Gorman.

> PMIS is composed of four interrelated subsystems. The data bases, the focal point of the PMIS design, are contained in the Data Base Subsystem. They are created and maintained through procedures in the Data Acquisition Subsystem, and are utilized in the Reports Subsystem which is the retrieval mechanism for PMIS. The final subsystem, Operational Control, provides an overall monitoring function to PMIS with its output as a complete log of PMIS activities. (p. 84)

PMIS also "provides an advanced tool based on technology to assist local school districts improve their efforts in long range planning, comprehensive evaluations, management oriented functions and general reporting" (Fiddleman and Gorman, p. 89).

COLPAS, College Planning and Analysis System, was developed by the College of Business Administration at the University of Houston to aid its planning activities. This system's primary components are the administrative and academic information systems.

One subsystem is the Undergraduate Student Information System. The system is

> To be used as a course advising tool on a semester-by-semester basis,
> To prepare the official degree plan for each student which would show him the courses for which he had been given credit and those which were required to complete his degree, and
> To be able to certify that a student had completed all course requirements for graduation and had the proper grade point average overall in his major field. (Vitulli and Singleton, 1972, p. 496)

Many other colleges have developed management information systems. A system which is somewhat different was developed by the School of Social Work at Michigan State University using the UDEANS method. This method "uses a computer-based information system to store data relevant to resource management and prepare useful reports for decision making by the community, the faculty, and the students" (Vinsonhaler et al., 1971, p. 33). It is unique since it maintains attitudinal data for faculty and students; and provides management reports to students as well as faculty and administrators. One management report of particular interest is the Student Program Guide which consists of three parts:

> first, a report is automatically produced if a student is not adequately meeting his program; the second and third parts of the Student Program Guide are courses and faculty ordered by interests in common with the student. These reports aid a student in planning his program of study according to his interests. (Vinsonhaler et al., p. 34)

A program to encourage the development of management information systems within institutions of higher education was undertaken by the Western Interstate Commission for Higher Education (WICHE). This program

> called for the development and implementation of compatible data bases including standard data elements and standard procedures, formulas, and models, in order to facilitate the exchange of comparable information among institutions of higher education and, further, to improve reporting to the state and federal levels. (Lawrence, 1969, p. 110)

In addition, the program was also concerned with the collection, storage, retrieval, and analysis of data in various operating systems such as the student information system, the accounting system, and the payroll system.

"In order to recognize the national character of the program and to stablize the funding of its activities" (Lawrence and Gulko, 1972, p. 45), the National Center for Higher Education Management Systems (NCHEMS), which is housed within the Western Interstate Commission for Higher Education, was developed in mid 1971. As described by Lawrence and Gulko,

> The purpose of NCHEMS is to promote the development of various aids to more knowledgeable resource allocation within institutions and agencies of higher education. Its activities are concerned with establishing a preliminary set of common data elements, program elements, a program classification structure, and program measures. The Center is also concerned with designing and implementing compatible management systems that will provide uniform data from which the cost of instructional programs by level of course, level of student, and field of study can be derived. (p. 49)

The preceding literature documents that management information systems have been developed to facilitate educational programs at all levels. Notably lacking from this discussion has been the development of such systems in colleges of education.

Since the late 1960s there has been a "national movement toward competency based instruction and individualization in teacher training courses" (Hayman and Mable, 1974, p. 60). Numerous institutions which have implemented such courses have developed management information systems to provide backup support for vital decision making.

The ComField model for teacher education developed by the Northwest Regional Educational Laboratory "specifies that 'instructional systems' will be employed to bring about professional competencies and their personalization; that instruction within these systems will be individualized. . .; and that a computer based management information system will be used to handle the frequent and diverse demands upon information" (Schalock, 1968, p. 2). The management information system consists of three logically interrelated files; the staff resource file, the student file, and the instructional system file (Coffin, 1968). It is capable of handling the following tasks:

1. Collect and store information on student performance and student expectancies.
2. Provide the instructional staff with information displays that identify students who are having trouble.
3. Order information about student needs so that instructional personnel can devote time to the most pressing problems.
4. Develop real time information for scheduling resources on a day-to-day basis so that they can be used in the most efficient manner.
5. Store and compile instructional system data for program effectiveness, appropriateness, and impact.
6. Combine information on student selection and retention characteristics.
7. Compile data about existing and planned acquisitions of resources to determine short- and long-range needs.

(Hamreus, 1968, p. 120)

Adoption of a competency based program by vocational and applied arts education at Wayne State University provided impetus for the development of a management information system. The "purpose of the VAE Management Information System is to select, store, process and transmit information to the faculty and students at a time when it can most effectively be used, enabling them to make more knowledgeable decisions" (Neuhauser, 1974, p. 1).

The data bank is composed of four computerzied files. The Master Student File consists of data obtained by computer from the university registration tapes. Data about each objective in the program are contained in the Course Objective and Parameter File. The Program Evaluation File contains a record on the completion of objectives by each student. Upon graduation, data from the students' Master Student File and Program Evaluation File are entered into the Follow-up File (Neuhauser, 1975).

Since "successful management of a CBTE program depends upon the regular collection of data from all those affected by the program" (Gentry and Jurs, p. 7), the computerized assessment and revision component of the CBTE program at the University of Toledo was developed.

The purposes of this component are to provide: a permanent record of individual performance of CBTE objectives, a record of the performance of groups of students on each objective and evaluation data.

Feedback from the system is provided to administrators, faculty, and students. Each student's advisor receives a report at least once a week which is used to determine recycling strategy and individualized sequencing. When a sufficient number of students have completed a module, a module revision report is generated. Such data give an indication of the effectiveness of instruction.

At Illinois State University the Surveillance system keeps track of students as they progress through the self-paced, competency based program in secondary education. Each student receives a weekly printout which serves as an official record of their progress in the completion of self-instructional packages. From this printout advisors are "able to detect any emerging pattern which can be altered to facilitate the student's learning progress" (Lorber, 1973, p. 6).

Data provided from other programs serve as a basis for decisions concerning the rewriting of packages and revision of merit weights and learning activities. Additional data such as the names of students who have completed all required packages prior to student teaching and the number of advisees assigned to each advisor are also provided.

The preceding systems were chosen as a representative sample of management information systems developed for CBTE programs. Of course, management information systems have also been developed for traditional time-based teacher education programs.

At Florida State University "the large number of students, programs and academic indicators which must be monitored, when coupled with a shrinking resource base and increased demands for 'accountability'" (Bonar and Dick, 1974, p. 42) necessitated the development of the Student Information Management System (SIMS). Rebhun and others (1974) reported a primary function of the system as

> that of providing academic performance data to academic advisors for students participating in the undergraduate teacher education program. The system incorporates an "early warning system" for identifying potential academic problems. Various quarterly student progress reports are distributed to advisors. Program administrators receive a number of reports summarizing computer checks of student records and progress for initial and continuing adherence to pre-set program standards. Secondary and tertiary functions of the system are the analysis of course demand to facilitate improved resource allocations and the structured storing of sufficient student biographical and performance data to actuate institutional research in teacher education. (p. 20)

It is possible to obtain a feeling for the breadth of this system by examining reports which are generated. Included among these reports are the Entrance Screening Report, the Biographical and Academic History Report, the Quarterly Progress Report for each student, the Course Enrollment Projection Report, the Preliminary and Final Student Teaching Screening Reports, and the Preliminary and Final Graduation Screening Reports.

Humanistic Field-based Teacher Education Programs

CEP began as an experimental program in the winter of 1969. The foundations for this program were laid at least as early as 1958 when Combs and Soper undertook a series of research studies on the characteristics of effective practitioners in the helping professions (Combs et al., 1969). This early research lead to a series of studies on teacher effectiveness at the University of Florida under the leadership of Arthur

Combs and at the University of Northern Colorado under the leadership of Richard Usher (Usher and Hanke, 1971). From these studies Combs developed a theory of teacher education which was published in 1965 in The Professional Education of Teachers: A Perceptual View of Teacher Preparation.

Wass and others (1974) succinctly outlined the development of CEP to its current status at the University of Florida as follows:

> a. It began in twelve years of basic research on the nature of the helping professions, especially on the nature of good and poor teachers.
> b. These research results were then combined with modern thinking from perceptual-humanistic psychology to formulate a theory of teacher education.
> c. This theory was given practical expression in an experimental program designed and placed in operation side by side with a traditional one.
> d. A program of research was then instituted to provide information concerning the relative effectiveness of the program. And finally,
> e. the experimental program was adopted by the department and over a two-year period replaced the old program.
> (Preface)

The major assumptions on which the program was based are that "a person learns best when: (1) learning is made personally meaningful and relevant, (2) learning is adjusted to the rate and needs of the individual, (3) there is a great deal of self-direction, and (4) there is a close relationship between theory and practice" (CEP, 1973, p. 1).

The program, which has been described in many articles and inhouse materials (Combs et al., 1965; Blume, 1971; Busby et al., 1972; CEP, 1973), consists of three main parts: seminar, substantive panel and field experience.

Activities in the program center around the seminar. The seminar provides for individual evaluation, counseling, guidance, and discussion about educational theory and practice. Students remain in the same seminar for their entire stay in the program. Because of the strong bonds built with one's seminar leader and fellow students, it is here that their education becomes most humanistic and personal.

The substantive panel consists of faculty members from different subject areas. Rather than completing courses in these areas, students must demonstrate competence in each area by completing learning activities designed in part by the faculty and in part by students. Certain activities are required while others are optional. Students may negotiate to replace both required and optional activities by competencies of their own design.

Upon entrance in the program, students are continuously involved in some type of field experience. As they progress through the various levels of field experience, their responsibilities and time spent in the school increase.

A recent study concerning CEP by Webb and Guinagh used a questionnaire, which was administered to two hundred and four students from two teams, to examine various aspects of the program. Webb and Guinagh (1975) cited the following strengths of the program:

1. The loyalty and enthusiasm the CEP program engenders in students, especially those who fall on the most positive end of the satisfaction continuum.
2. The fact that the 81.4 percent of students indicated a growth in their sense of responsibility since joining the program.
3. The high degree of student satisfaction regarding seminars.
4. The general belief among students that learning activities and substantive areas are doing an adequate to outstanding job in preparing them for a career in teaching.
5. The near universal enthusiasm regarding the field experience.
6. The impressive number of students indicating that many substantive areas and the CEP experience in general have helped them to develop a positive self concept.

7. The fact that the amount of student cheating in CEP seems to be substantially lower than is found in other college programs.
8. The fact that students seem to spend a good deal of time learning from and with one another.
9. The fact that most students believe they have the power to make changes in CEP. (p. 50-51)

In addition, the following weaknesses of the program were cited:

1. The fact that so few students engage in learning activities out of what they define as "a need to know."
2. The fact that over half of the students interviewed indicated that they spent three hours or less per week in contact with faculty on substantive area work.
3. The apparent lack of connection between field work and substantive area concerns.
4. The generally inadequate way in which students are introduced into the Childhood Education Program.
5. The failure of community sessions to aid in developing a sense of community.
6. The fact that students tend to prefer a more traditional program of teacher preparation which incorporates work in the schools.
7. The large percentage of students who indicate uncertainty as to whether or not the faculty is doing an adequate job and who indicate that it is difficult to locate and talk with professors. (Webb and Guinagh, p. 50)

These findings indicate that there is room for improvement in the program.

Many facts about the program which are relevant to the current study were presented in a recent discussion of CEP as an alternative approach to CBTE by Combs and Kinzer (1974). Since the program is "constantly involved in an ongoing experiment in curriculum development and teacher education" (Combs and Kinzer, p. 188) it cannot be rigid. Combs and Kinzer stated:

> Faculty and students together are involved in a continual search for ways to improve (1) our data collection procedures; (2) our program organization; (3) collaboration between public schools and universities; and (4) our identification of program goals, implementation, and evaluation. (p. 188)

Faculty and students also "fear that increased specificity or explicitness will bring about rigidity or at least reduced flexibility" (Combs and Kinzer, p. 188) in the program.

The preceding review of the literature has centered on the areas of management information systems and humanistic field-based teacher education programs. Attention is now focused on the management system which is currently used by CEP.

Description of the Current Management System

The current management system consists of four major parts: the team office, the seminar leaders, the substantive panel members, and the Undergraduate Office in the College of Education. Since each part maintains separate records on students, duplication of data exists between and among the various parts.

The team office processes all students who are in the team. Initially, students are assigned to the team by the Undergraduate Office. During the week prior to the beginning of the quarter, the team leader assigns each new student to a seminar where with rare exception he is to remain for his entire stay in the program. Within this same time period, this office must provide academic counseling for new students. Consequently when there are a large number of new students, the situation in the team office becomes rather hectic.

At the beginning of each quarter, seminar leaders send the names and local addresses of the members of their seminar to the team office. This information is used to produce the team list and student roster which are made available to all members of the team.

Other activities are coordinated by the team office. For instance, it supervises the effort in checking that upcoming graduates have met all program requirements and students going into their intensive have fewer than six learning activities to complete.

Seminar leaders maintain detailed records on the progress of their students in terms of their basic personal data, registrations, learning activities, and field and community experiences. It is not unusual to find that these records are not up to date. Learning activities may be missing, registrations may be either missing or not updated, grades from required courses may not be recorded, and field and community experience evaluations may also be missing.

In the seminar which was observed, part of the problem associated with unrecorded learning activities may be related to the policy of holding students responsible for recording completed activities on their program record. All learning activity slips except for the final evaluations were not to be filed in the seminar office, rather students were instructed to keep them at home. At the end of each quarter when student progress must be assessed, an inaccurate record of what has been completed may create difficulties.

Incorrect registrations are probably caused by failure to change the program record when a student either drops or adds a course. This may result in difficulties when checking both the completion of required courses and the number of credit hours needed for graduation. Fortunately, seminar leaders may use the student's permanent record from the Undergraduate Office to update missing or inaccurate data, but this is a rather time consuming operation. At periodic intervals seminar leaders are requested to update the planned program, which is maintained by this office, for each of their students.

Each substantive panel member maintains a record for each of his students. There is a wide variety in the amount and kinds of data which are contained in these records. Some faculty merely have a file card with the student's name, the activity number and title, the date the activity was completed, and perhaps a few comments. Other faculty members maintain a detailed record of the workshops and help sessions attended, all learning activities which are submitted, and general information about the student's academic progress. Still other faculty members retain copies of all learning activity slips and general comments about student progress. Although detailed records are helpful to the faculty, they are quite time consuming to maintain. Consequently, the type of record-keeping system which is used is a matter of personal preference and efficiency. Unfortunately, brief and possibly inaccurate faculty records of student progress may result in problems since students work at their own rate and the requirements have a tendency to change from quarter to quarter and instructor to instructor.

Most substantive panel members have little knowledge about their student's progress in other areas. Such limited knowledge may be an obstacle in working with students. The only way to obtain valid information about a student's progress in other areas is to contact the seminar leader or other substantive panel members.

Learning activities are submitted to the correct substantive panel member. At times, faculty members in one substantive area work together. Consequently, different faculty members may be assessing one student's progress. In most cases assessed activities are sent to a central location where students obtain the learning activity and the learning activity slip. If the learning activity is acceptable, the student must then go to his seminar leader's office and record it in his program record. Some seminar leaders want all of the student's learning activity slips in their folders while others only want the final evaluations in each substantive area.

Seminar leaders rely on obtaining information from the Undergraduate Office which maintains a permanent record on each undergraduate student in the College of Education. The record-keeping system used by this office is quite complicated since the degree of completion of the personal data sheet and registration activity status determine where the student's permanent record may be found in the office. Sometimes these records are missing. Because of the tremendous demands placed on this office, grade slips from the previous quarter and current registration slips are not always promptly filed. Thus problems may arise when seminar leaders need readily accessible and accurate information about their students.

CHAPTER III

PROCEDURE

Overview

Van Dusseldorp (1969, 1971) defined the steps in the development of an information system. These steps were altered to meet the problem under consideration. As a result, the following steps were identified:

1. Specify the goals and objectives of the system.
2. Develop the system.
3. Test the system.
4. Revise the system.
5. Evaluate the system.

The development procedure is described in terms of these steps.

Specify the Goals and Objectives of the System

Perhaps the most difficult and important step in the development of the computer-based management information system was to specify the goals and objectives of the system. A basic understanding of the program was essential. Consequently an initial step was to review published and in house materials about CEP (e.g., Combs et al., 1965; Blume, 1971; Busby et al., 1972; CEP, 1973).

Subsequently the team leader of the team which was to serve as the model was contacted. General information was obtained about CEP, this team, and the current management system used by the team. Copies of the various forms used for student record keeping and reports generated by the team were also procured.

In order to obtain a more thorough understanding of CEP and the model team in particular, the various phases of the team's program were observed for one quarter. The investigator also participated whenever possible in both seminar and community session activities.

At the beginning of the observation period it became apparent that the program seemed to foster an open atmosphere in which the students felt free to discuss CEP. Thus, the students' likes, dislikes, concerns, and problems with the program became readily apparent.

During the later part of this observation period the other faculty members on the team were individually interviewed. In order to orient them to the problem at hand, a hypothetical computer-based management information system for the team was described. Faculty were informed of the importance of their input into the system being developed and asked to help in this development procedure. Seminar leaders were asked to describe the kinds of records they kept on their seminar, the problems associated with keeping these records, and other types of information they would like to have about their students. Substantive panel members were asked to describe their record-keeping practices, to identify the problems associated with keeping these records, to suggest other types of information they would like to have about students, and to specify the current requirements for learning activities in their substantive area.

A complete description of the current management system and the problems inherent to this system were discussed in Chapter II.

From the information obtained during the observation period, the goals and objectives of the management information system were defined as follows:

Goal:	The system will provide readily accessible and reliable information about the team and students in it.
Objectives:	The system will be capable of generating reports to serve the faculty in decision making about team policy and individual students.
	The system will keep track of student progress.
	The system will have a broad data base which can be used for research and program evaluation.
	The system will be capable of handling the fluid nature of CEP.

Develop the System

The actual development of the system was broken down into the following activities: specify in detail the data to be kept on each student, develop the software using PL/1, design the procedure for the collection and dissemination of information, and write directions on how to use the system.

An initial layout of a student record was constructed using the personal data sheet, program record, field experience evaluation and request forms, community experience evaluation form, learning activity slip, and faculty suggestions as a guide. The team leader and other individuals concerned with CEP made suggestions for changes in the layout. A revised layout, which proved satisfactory to all, was then constructed. This revised layout is the structure PERSONAL (Appendix C) which is included in each program in the system.

Most data in the student record were to be recorded in coded form. Uncoding was done by using dictionaries of the item names and associated codes. Some dictionaries which were developed were straightforward and easy to construct since there was a clearly defined set of possibilities.

Other dictionaries associated with the community experience, field experience, and learning activities were more difficult to construct since they were not clearly defined.

The more abstruse dictionaries were developed in several stages. First, old student records were examined to find various possibilities for each item. Sample dictionaries were then constructed and examined by the team office to correct ambiguous wording, eliminate duplicate entries, and include omissions. Later, these updated dictionaries were examined by other personnel in the team office for further corrections.

An attempt was made to make the length of data items not recorded in coded form compatible with the Data Element Dictionary for Student Information used by Florida's Community Colleges and the status record layout used by Administrative Computing Services at the University of Florida. In cases of disparity between the lengths, that length which more closely characterized the data in the current management system was chosen. Still other changes were made in light of typical data that existed in the current management system.

The software was written in PL/1 over a period of months. Each program in the system was to perform a specific function. Initial programs in the system were written and debugged solely using card input. The output generated by these programs was checked, altered if necessary, and finally approved by the team office for form and clarity. Once all of these initial programs were operating satisfactorily, they were converted to use tape input and output of student records.

The field experience request form was rather antiquated since it was designed using the student internship request form with an entirely different program as a model. Consequently, unnecessary information was included in the field experience request form. Also, descriptions of levels in CEP were incomplete and no longer applicable. The team could not change this form without the approval of the Student Teaching Office which serves as a liaison between the College of Education and local school systems. Personnel in this office felt the form should only be revised in a joint effort between school personnel, the college, and the team. Because of the local school schedule, this meeting was not held until a few weeks before the system test was to begin. Suggestions were made concerning what could be omitted from the current form. Using suggestions made at this meeting, the field request form was then revised. A program was then written to generate the new form.

In order to facilitate the entry of data into the system, it was necessary to redesign certain current forms such as the learning activity slip and field experience evaluation form. New forms were designed to enter initial data into the system and to request printouts using specific programs in the system.

A users' manual (Appendix D) was developed. For each program in the system there is a description of the following items: output description, class size, number of tape drives, DD statements, data set-up, data recording or request form if applicable, sample data deck, and sample output.

Test the System

A test of the system was conducted on a seminar of twenty-eight students for a period of one quarter. Of the students in this seminar nine students had been previously enrolled in the program; nineteen were new.

Prior to the test period as much data as possible, such as the student's permanent address, previous registrations, colleges attended, etc., were collected from the records of previously registered students. Once these data were obtained the student records were created and entered into the main file which was sorted into ascending order by social security number.

At the beginning of the quarter the previously registered students were asked to list the courses they were taking, their local addresses and phone numbers. In addition, they were requested to fill out a form about completed learning activities. For each completed activity they were to list the area, activity number, completion date, and faculty member. The main file was then updated with the new data items.

During the first week of the quarter new students were asked to list their basic personal data and courses they were taking. Records for the new students containing these data were created and added to the main file which was again sorted. Registration information was then added by updating the main file.

At the first team meeting of the quarter, the system was explained to the team, samples of the output from the system were shown to the faculty, and they were asked to use the revised learning activity slip in the seminar involved in the test. After a brief discussion, it was agreed that the revised slip would be used in that seminar.

The use of the computer-based management information system was explained to the members of the seminar involved in the test. Students were asked to help in the test of the system by using the revised learning activity slips. The new procedure for handling completed learning activities was then explained. Each student was given approximately twenty to thirty blank learning activity slips. They were instructed to attach one slip, with the upper portion filled out, to each learning activity as it is submitted.

Since the current management system could not be disrupted, it dictated how data on learning activities would enter the system. Substantive panel members sent the activity with the associated learning activity slip to a central location. Students picked up these activities at their convenience and filed the slips in the team office. During the later part of each day, student folders in the team office were checked. A summary was made of the data on all returned learning activity slips. The activity slips were then placed in the student's permanent folder. Cards for these updates were punched. The main file then was updated sequentially in ascending order by social security number. If no faculty requests were made, then these updates were held until the weekend. Thus, the system was minimally updated once a week. On a limited budget, it seemed wasteful to make a computer run each day if it was not needed.

Faculty members were shown sample output and given instruction in the use of the system. They were given a number of faculty request forms and a listing of the team members by seminar with their associated social security numbers. All faculty requests were filed in the team office. Computer runs for the faculty requests were made on a daily basis. Output was filed in each faculty member's mailbox early the next morning.

Each faculty member was urged to see what the system could do. Unfortunately, few faculty members attempted to use the system. Their own records were readily available. Because of the low response rate, a memo (Appendix B) was sent to each faculty member. Only twenty percent of the faculty responded to the memo.

Since the system was to be evaluated by each faculty member, it was felt that they must have some sample output. A hypothetical request was made and submitted for each faculty member. When the request with associated output was personally delivered to the correct individual, an appointment was made for the final evaluation. Each faculty member was urged to examine the output carefully and to ask questions.

Other programs in the system such as the local address list, team list, and graduation list were run at the appropriate time during the quarter. Output from these programs was given to the seminar leader.

Revise the System

System revisions were made during the test period. There were no major changes to the initial programs in the system, but new programs were added to the system.

At the beginning of the quarter it became apparent that the faculty did not have the social security numbers of their students. Consequently, one of the options on the faculty request form could not be used. A program was therefore written to print a list of team members by seminar with their associated social security numbers.

Revision of the field experience request form resulted in a change in the basic layout of the student record. This affected every program in the system since travel and work experiences were added to the structure PERSONAL. The only major revision was in the program which updates items in PERSONAL. This program was changed to include update routines for these new items.

After the field experience request form was revised, a program was written to generate the form printing specific data on each student. That information which had to be completed by the student, for instance the grade level they wanted to work with next quarter, was not included on this printout.

Once the system was in operation, it became apparent that basic data in the system could be easily updated. A program was written to print the basic personal data which tends to change. Students could be given a copy of this printout at the beginning of the quarter. Invalid information could be crossed out and corrected. This corrected printout could be returned to the seminar leader and used to update the student's record.

All of the data stored in PERSONAL were printed in some form by some program, yet no program in the system printed everything. Thus the program named ALL, which prints everything contained in a student's record, was written.

Evaluate the System

During a three-day period at the end of the quarter, the investigator met individually with each faculty member to evaluate the system. First each faculty member was asked if he had any questions about the output from the system.

An evaluation was then conducted using the CEP Management Information System Evaluation (Appendix A) which was developed prior to the test of the system. In addition to those direction on the evaluation form, the faculty were given the following instructions in order to give them a better frame of reference for answering certain items.

1. For the first item, the faculty were asked to assume that they used all of the information on students.

2. For the third item, the faculty were asked to assume that the computer-based management information system had replaced the current management system and provides them with requested information. How would this affect their use of time?

CHAPTER IV

DESCRIPTION AND EVALUATION OF THE COMPUTER-BASED MANAGEMENT INFORMATION SYSTEM AS TESTED

Description

Chapter III contains a description of the basic characteristics of the computer-based management information system which was tested. Missing from this description were the effect of the system on the faculty, the effect of the system on decision making, and the problems of acceptance by students and faculty.

If the system were in operation for an entire team, certain clerical activities performed by the team office would be eliminated. The compilation of the team list, local address list, and graduation list would be completed merely by requesting the execution of certain programs.

Seminar leaders are provided with timely information on individual students or their entire seminar. Background information which currently must be obtained from the Undergraduate Office is now easily accessible. In addition to data in the current system, summary information may be obtained about a student's learning activities and field experiences. This information, which provides a concise view of a student's overall progress, growth, and needs, can be valuable in academic and personal counseling. Since most data on the field experience request form can be automatically generated, time will not be wasted in seminar meetings completing this form. Other questionnaires may be eliminated by a readily accessible data base.

The effect of the system on substantive panel members is substantial, since they have more information concerning the team, a seminar, or an individual student, and the information is better organized and readily available. Advance information about a student's overall progress in the program can be used for counseling. It may also prompt faculty members to discuss a particular student's progress or potential. With more information available about a student, it will be easier to plan individual learning activities.

Accurate and readily accessible data on student progress in CEP may be used for program evaluation and research. Perhaps certain questions about the program can now be answered.

Students were concerned about the use of a computer in a humanistic program. Since many students were new, the description of what the system could do for them was of little meaning. It was just an additional problem with which students had to contend. With the mass of students on campus where one is often identified by merely a social security number and no strong ties already established in CEP, this concern was quite understandable.

Previously registered students also expressed concern. Since their records were not up-to-date, they were asked to list those learning activities which they had completed. Knowing of the systems' capabilities, this posed a threat. Suddenly they realized that the faculty would know more about their total progress. They wondered if faculty members would spend less time with them. What effect would this really have on them?

Student reluctance to the use of the system probably affected their cooperation in the test. This reluctance may account for the fact that often they simply forgot to attach the revised learning activity slip to activities which they submitted. In addition, many previously registered students procrastinated and did not return the list of learning activities which they had completed until the middle of the quarter or later.

Although the faculty recognized the potential benefits of the system in their evaluation of it, they were reluctant to use it. Some of this reluctance may stem from the fact that, according to the procedure for the test of the system, they had to maintain their own set of records which they were accustomed to using. Since they were required to maintain these records, could they really trust the system? The mere idea that such a system existed was difficult to comprehend. With only part of the team involved in the test, the adjustment was understandably difficult.

Evaluation

The evaluation of the computer-based management information system was described in Chapter III. Since the number of faculty responses to the CEP Management Information System Evaluation was small, no statistical tests were performed. Results are summarized by frequency and percent for each part of the five items in Tables 1 through 5. A collection of some open-ended responses, given by the faculty to the question why they would or would not like to see the system in operation for the entire team, is presented in terms of the five items.

In item one faculty were asked whether the system which was developed provided sufficient information about the students. One faculty member would like to see the system in operation for the entire team because it provided "overall background information such as: addresses, graduation dates (potential), number of learning activities completed in special areas for me - also overall, and the number of terms registered in the program." Another faculty member said the system would "offer concise, updated data on student progress through the program."

Item two was concerned with the overall evaluation of the system. One faculty member was impressed by the "quick turn-around on getting point-in-time status measures on individual student performance" which would help in being "more effective in my monitoring and guidance roles." Another faculty member found the system to be "an efficient means of organizing information and providing a manageable procedure of record keeping for all team students."

The third item was designed to assess how the use of the computer-based management information system, as the only source of data on students, would affect the time faculty spent with students in various situations. One faculty member thought the system "would make management more efficient by encompassing all areas." Another said "the information provided will aid me in organizing my perspective of the individual student in and out of my particular field; enabling me to spend more time effectively and efficiently in dealing with them."

Item four was designed to assess faculty attitudes about the effect of the system on their evaluation of students, counseling, and planning. According to faculty responses, the system would "help in planning certain curricular alternatives for students" and would permit the "substitution of professional judgments of a student for a letter grade when an estimate of quality is needed."

In item five the faculty members were asked if they would like to see the management information system in operation for all of Team III. One faculty member said "it may indeed cause one to have renewed appreciation of the wide variety of operations existing within the program."

Another indicated that in the system "there is a rich untapped potential that alone will provided better answers and will enhance my role." In addition the system provides "more and better opportunities for sideways communication." Still another faculty member would like to see the system in operation "for my own organization and a happier life."

A complete tabulation of responses to the amount of information provided by the system in various areas is presented in Table 1. These data indicate that the system did provide at least a sufficient amount of information for the areas which were examined. A few responses of "more than needed" were recorded. These responses indicate that there is a divergence of opinion, however slight, on what types of information faculty members need.

Responses to the item concerning the performance of the system are presented in Table 2. Faculty judged the performance of the system to be either satisfactory or excellent.

In Table 3 a tabulation of responses to how the system would affect the amount of time spent with students in various situations is presented. Faculty members felt they would spend about the same amount of time with students. Responses of "less time" and "more time" probably are related between and among all parts of this item. Faculty would need to spend less time in obtaining information, thus there would be more time available to interact with students in this or other situations.

Data which are presented in Table 4 indicate the effect of the system would be either positive or nondetrimental to the program. Faculty who have not been on a substantive panel did not respond to part 4A.

Faculty responses to the fifth item are presented in Table 5. All

TABLE 1

CEP MANAGEMENT INFORMATION SYSTEM EVALUATION
RESPONSES TO ITEM 1

The management information system was developed to provide faculty members on the team with timely and accurate information on students. Does the system as developed provide sufficient information for your purposes in the following areas?

Item	Frequency of Responses		
	Not Sufficient	Sufficient	More than Needed
A. List of students in each seminar	0	11	1
% of Respondents	0.00	91.66	8.33
B. Addresses of students in each seminar	0	11	1
% of Respondents	0.00	91.66	8.33
C. Listing by seminar of upcoming graduates with missing learning activities in each area	0	11	1
% of Respondents	0.00	91.66	8.33
D. Student's profile on the Myers Briggs Type Indicator	1	10	1
% of Respondents	8.33	83.33	8.33
E. Student's completion of required courses	0	11	1
% of Respondents	0.00	91.66	8.33
F. Student's registration by quarter and grades received	0	11	1
% of Respondents	0.00	91.66	8.33
G. Student's learning activities completed to date	0	12	0
% of Respondents	0.00	100.00	0.00
H. Summary of student's learning activities completed to date	0	12	0
% of Respondents	0.00	100.00	0.00
I. Student's field experiences completed to date	0	11	1
% of Respondents	0.00	91.66	8.33

TABLE 1 - Continued

Item	Frequency of Responses Not Sufficient	Sufficient	More than Needed
J. Summary of student's field experiences	0	11	1
% of Respondents	0.00	91.66	8.33
K. Student's community experience	0	11	1
% of Respondents	0.00	91.66	8.33

TABLE 2

CEP MANAGEMENT INFORMATION SYSTEM EVALUATION
RESPONSES TO ITEM 2

Item	Frequency of Responses Poor	 Satisfactory	 Excellent
Considering all aspects of the management information system, in your opinion, how well did the system perform?	0	5	7
% of Respondents	0.00	41.66	58.33

TABLE 3

CEP MANAGEMENT INFORMATION SYSTEM EVALUATION
RESPONSES TO ITEM 3

Compared to the amount of time you now spend, how much time do you think you would spend with students in the following situations if the management information system was your only source of data on students?

Item	Frequency of Responses		
	Less Time	Same Amount of Time	More Time
A. Individual discussion	0	10	2
% of Respondents	0.00	83.33	16.66
B. Individual counseling (academic - providing evaluative feedback)	2	6	4
% of Respondents	16.66	50.00	33.33
C. Individual counseling (personal)	1	7	4
% of Respondents	8.33	58.33	33.33
D. Small group (e.g. seminar, discussion group) meetings for instruction and planning	1	9	2
% of Respondents	8.33	75.00	16.66
E. Large group (e.g. workshop, etc.) meetings for instruction and planning	1	10	1
% of Respondents	8.33	83.33	8.33

TABLE 4

CEP MANAGEMENT INFORMATION SYSTEM EVALUATION
RESPONSES TO ITEM 4

Assume that the management information system has replaced the manual record keeping system.

Item	Frequency of Responses	
	Yes	No
A. Would you make fewer comments on student's learning activities than you did before the management information system was installed?*	2	8
% of Respondents	16.66	66.66
B. Would you do a better job counseling (working with) students because you have more readily accessible data on their progress in the program?	12	0
% of Respondents	100.00	0.00
C. Would you have more data which could be used for better evaluative judgment?	11	1
% of Respondents	91.66	8.33
D. Would you find it easier to plan individual learning activities?	8	4
% of Respondents	66.66	33.33

---

*The total responses differs from the number of persons interviewed due to the fact that respondents occasionally gave no response to this item.

TABLE 5

CEP MANAGEMENT INFORMATION SYSTEM EVALUATION
RESPONSES TO ITEM 5

Item	Frequency of Responses	
	Yes	No
Would you like to see the management information system in operation for all of Team III?	12	0
% of Respondents	100.00	0.00

faculty members indicated they would like to see the system in operation for the entire team. In response to the open-ended question, faculty members indicated different reasons for this affirmative response.

CHAPTER V

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS FOR FUTURE RESEARCH AND DEVELOPMENT

Summary

This study examined the feasibility of developing and operating a computer-based management information system for CEP to provide readily accessible and reliable information about the team and students in it.

A basic understanding of the program was essential. Initially, materials concerning CEP were reviewed. Then the team which was to serve as the model during the test was observed for a period of one quarter. In addition, faculty on this team were interviewed. From the information which was obtained, the goals and objectives were stated.

Actual development of the system was broken down into the following activities: data to be maintained on each student were specified in detail, system software was developed in PL/1 over a period of three months, procedures for the collection and dissemination of information were established, and a users' manual was written.

A test of the system was conducted for a period of one quarter on a seminar of twenty-eight students and the faculty members on their team. The computer-based system was operated parallel with the current management system during the test period. This type of test provided checkup and backup capabilities.

During the test period revisions were made to the system. Although there were no major changes in the initial programs in the system, new programs, each which performed a specific function, were added when their need became apparent.

At the end of the quarter, faculty evaluation of the system was conducted on a one-to-one basis using the CEP Management Information System Evaluation form. This form, which was developed prior to the test of the system, consisted of both structured and open-ended responses.

Since the number of faculty responses to the evaluation instrument was small, no statistical tests were performed. Results were summarized by frequency and percent for each part of the five items.

All faculty members would like to see the system in operation for the entire team since it provided timely and accurate information about the students and the team. This information could be used for counseling, decision making, and research.

Although faculty members would like to see the system in operation for the entire team, they were reluctant to use it despite prodding from the investigator. This reluctance probably stems from a general fear about the use of the computer and the limitations imposed by the test. Hopefully this reluctance can be eliminated by further education during full-scale implementation.

Conclusions

This study was concerned with the feasibility of developing and operating a computer-based management information system for CEP. From the results, it is concluded that a management information system can be developed. However, faculty reluctance to use the system creates a question as to the operational feasibility of such a system in the program at this time.

The system performed well in that it provided timely and accurate information about students and the team. This information could be used for counseling, decision making, and research.

Faculty reluctance to use the system is attributed to the following factors: only some students in the team were involved in the test; the test while of limited duration with no promise of further commitment posed a burden on the user by being something new with which to contend; records had to be maintained for all students during the test period; adjustment to something which is new takes time; and difficulty in accepting the use of the computer in CEP did exist.

A definite problem exists in overcoming faculty reluctance to the use of the computer in CEP. This reluctance probably stems from previous personal experience and knowledge of problems other people have had with the computer. It is quite possible that faculty members are not aware they are afraid, yet their actions indicate to the investigator that they may well be.

Hopefully this reluctance can be overcome through further association. Faculty must become aware that the system is not a menace; it is there to help and support them.

CEP faculty members are not heavily data oriented. In addition, the program allows considerable latitude while not pressing for the gathering of data. Consequently, definite problems did exist in obtaining missing data on previously registered students. It was difficult to impress upon faculty the importance of supplying the system with timely and accurate data. Without such data, the value of any system is questionable.

Everyone involved learned that the establishment of a computer-based management information system takes a great deal of time and patience.

Recommendations for Future Research and Development

Constraints imposed by the university, model team, computer center, and budget limited the computer-based management information system which was developed and tested. Assume that these constraints are removed and a hypothetical system for CEP is developed. In this hypothetical system the data which are currently maintained and reports which are currently generated by the computer-based management information system would not be changed. Alterations would be seen in the flow of data into the system, the number of reports which are generated, the contents of various dictionaries, and faculty orientation to use of the system.

All of the basic data and registration information could conceivably be obtained from Academic Computing Services at the university on magnetic tape. Records maintained by this service, which are updated at the beginning and end of each quarter by adding and deleting courses, adding and changing grades and changing the appropriate basic profile data, could be used to update the main file. Unfortunately, these records are not in a form which permits an update tape to be generated without considerable effort. If an update tape could be more easily generated, then another program capable of handling these update records would have to be written.

The use of data from Academic Computing Services would eliminate a lot of effort in gathering data which has already been collected in another office on campus. However a problem would be the length of time before the data could be made available to CEP.

If the computer-based management information system is to replace individual substantive panel members' records, it should be capable of maintaining data on all learning activities which are submitted. A learning activity slip could be constructed to allow constructive criticism to be made for those activities which are not acceptable. Whenever an activity is updated the new comments could replace the old comments. If a record were to be kept of the number of times an activity is returned, it could be used for program evaluation, program revision, and research.

In order to have the data on learning activities up to date, the current information flow of data on learning activity slips into the system would have to be changed. Learning activities would be collected from the faculty on a daily basis. These data on the learning activity slips could then be immediately entered into the system. Students could still obtain the completed activities and attached learning activity slips at a central location.

Substantive faculty members could receive a weekly listing of their students and the activities they have submitted in that substantive area. Additional information could be provided whenever a request is made.

Students could periodically be given a report of their progress. They could also be allowed to request information on their own progress at any point in time.

In an effort to reduce faculty reluctance to use of the system, a formal faculty discussion about the system and its potential could be held. Such discussion about the system could create both interest and curiosity. Thus the faculty could in effect "sell" the use of the system to one another.

The problem of faculty reluctance to use of the system has convinced the investigator of the need for research on how to most effectively interface humans with sophisticated technological systems. Apparently we do not know a great deal, as yet, concerning how to enable human beings to identify with and invest in a systematic management approach to instruction.

As the computer-based management information system is used, the faculty would hopefully begin to see other uses of the system which may require the generation of new reports. Additional data items might also be desired. Before entering new data items into the system, they should be judged on their potential use in decision making, research, etc.. The dynamic nature of the system would readily allow for the addition and deletion of data items when necessary.

As with any other innovative program, there are many unanswered questions about CEP. The computer-based management information system will provide a data base from which many necessary and interesting studies can be conducted. Hopefully these studies will be realized.

APPENDIX A

CEP MANAGEMENT INFORMATION SYSTEM EVALUATION

CEP
MANAGEMENT INFORMATION SYSTEM
EVALUATION

DIRECTIONS: PLEASE INDICATE YOUR CHOICE BY CHECKING THE APPROPRIATE RESPONSE.

I. The management information system was developed to provide faculty members on the team with timely and accurate information on students. Does the system as developed provide sufficient information for your purposes in the following areas?

A. List of students in each seminar	____Not sufficient ____Sufficient ____More than needed
B. *Addresses of students in each seminar	____Not sufficient ____Sufficient ____More than needed
C. *Listing by seminar of upcoming graduates with missing learning activities in each area	____Not sufficient ____Sufficient ____More than needed
D. Student's profile on the Myers Briggs Type Indicator	____Not sufficient ____Sufficient ____More than needed
E. Student's completion of required courses	____Not sufficient ____Sufficient ____More than needed
F. Student's registration by quarter and grades received	____Not sufficient ____Sufficient ____More than needed
G. Student's learning activities completed to date	____Not sufficient ____Sufficient ____More than needed
H. Summary of student's learning activities completed to date	____Not sufficient ____Sufficient ____More than needed

---

*These things are furnished by the system although you may not receive them in the form of computer printouts.

I. Student's field experiences completed to date

_____Not sufficient
_____Sufficient
_____More than needed

J. Summary of student's field experience

_____Not sufficient
_____Sufficient
_____More than needed

K. Student's community experience

_____Not sufficient
_____Sufficient
_____More than needed

II. Considering all aspects of the management information system, in your opinion, how well did the system perform?

_____Poor
_____Satisfactory
_____Excellent

III. Compared to the amount of time you now spend, how much time do you think you would spend with students in the following situations if the management information system was your only source of data on students?

A. Individual discussion

_____Less time
_____Same amount of time
_____More time

B. Individual counseling (academic - providing evaluative feedback)

_____Less time
_____Same amount of time
_____More time

C. Individual counseling (personal)

_____Less time
_____Same amount of time
_____More time

D. Small group (e.g. seminar, discussion group) meetings for instruction and planning

_____Less time
_____Same amount of time
_____More time

E. Large group (e.g. workshop, etc.) meetings for instruction and planning

_____Less time
_____Same amount of time
_____More time

IV. Assume that the management information system has replaced the manual record keeping system.

A. Would you make fewer comments on student's learning activities than you did before the management information system was installed?

_____Yes
_____No

B. Would you do a better job counseling (working with) students because you have more readily accessible data on their progress in the program?

_____Yes
_____No

C. Would you have more data which could be used for better evaluative judgments?

_____Yes
_____No

D. Would you find it easier to plan individual learning activities?

_____Yes
_____No

V. Would you like to see the management information system in operation for all of Team III?

_____Yes
_____No

Why?

APPENDIX B

MEMO TO FACULTY MEMBERS

DATE: Nov. 18, 1974

MEMO FROM: Bernice Doerr

TO: All CEP3 faculty members

SUBJECT: At the end of the quarter you will be asked to evaluate the management information system. If you have not yet used the system please try to do so in the next few weeks.

Thank you.

APPENDIX C

PROGRAMS IN THE COMPUTER-BASED MANAGEMENT INFORMATION SYSTEM

Program	Page
ALL	63
BASIC	81
BUILD	86
CHECK	89
FILE	94
GRADS	96
LOCAL	99
PRINT	102
PURGE	110
REQUEST	112
SS	115
TEAM	118
UPDATE	121

```
ALL:    PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:        ALL

        OUTPUT
        DESCRIPTION:    LISTING OF ALL DATA ON STUDENTS AS
                        REQUESTED BY FACULTY MEMBERS.

        ALGORITHM:      READ THE INITIAL DATA.  OBTAIN A FACULTY
                        REQUEST.  IF THE REQUEST TYPE IS EQUAL TO
                        ONE. PRINT ALL INFORMATION IN THE RECORDS
                        OF THE MEMBERS OF THE SPECIFIED SEMINAR.
                        OTHERWISE. PRINT ALL INFORMATION IN THE
                        RECORDS WITH THE SPECIFIED SOCIAL
                        SECURITY NUMBERS FOLLOWING THE REQUEST
                        UNTIL AN '*', WHICH INDICATES THE END OF
                        THIS REQUEST, IS READ.  CONTINUE TO
                        PROCESS THE REMAINING REQUESTS IN THE
                        SAME MANNER.
                                                              */
        /*

        READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
        USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
        FACULTY_NO.
                                                              */
        GET FILE(SYSIN) EDIT(MIN,MAX)(COL(1),2 (F(1),X(4)));
        BEGIN;
ONE:    DECLARE
          /*
          EACH STUDENT'S RECORD CONSISTS OF PERSONAL AND
          ACADEMIC DATA.                                      */
          1 PERSONAL,
            /*
            SOCIAL SECURITY NUMBER                            */
            2 SOC_SEC_NO CHARACTER(9),
            /*
            NAME                                              */
            2 NAME,
              /*
              LAST NAME                                       */
              3 LAST CHARACTER(14),
              /*
              FIRST NAME                                      */
              3 FIRST CHARACTER(10),
              /*
              MIDDLE NAME                                     */
              3 MIDDLE CHARACTER(14),
            /*
            PERMANENT ADDRESS                                 */
            2 PERM_ADDR,
              /*
              STREET                                          */
              3 STREET CHARACTER(20),
              /*
              CITY                                            */
              3 CITY CHARACTER(20),
              /*
              POSTAL ABBREVIATION FOR THE STATE               */
              3 STATE CHARACTER(2),
              /*
              ZIP CODE                                        */
              3 ZIP CHARACTER(5),
            /*
            PERMANENT PHONE NUMBER INCLUDING AREA CODE        */
            2 PERM_PHONE CHARACTER(10),
            /*
            LOCAL ADDRESS                                     */
            2 LOCAL_ADDR,
              /*
              STREET                                          */
              3 STREET CHARACTER(20),
              /*
              CITY                                            */
              3 CITY CHARACTER(20),
              /*
              POSTAL ABBREVIATION FOR THE STATE               */
              3 STATE CHARACTER(2),
              /*
              ZIP CODE                                        */
              3 ZIP CHARACTER(5),
            /*
            LOCAL PHONE NUMBER                                */
```

```
2 LOCAL_PHONE CHARACTER(7),
/*
SEX: ONE CHARACTER CODE                                  */
2 SEX CHARACTER(1),
/*
DATE OF BIRTH                                            */
2 BIRTH_DATE,
  /*
  MONTH OF BIRTH: TWO DIGIT CODE                         */
  3 MONTH FIXED BINARY,
  /*
  DAY OF BIRTH                                           */
  3 DAY FIXED BINARY,
  /*
  YEAR OF BIRTH: LAST TWO DIGITS OF THE YEAR             */
  3 YR FIXED BINARY,
/*
MARITAL STATUS: ONE DIGIT CODE                           */
2 MARITAL_STAT FIXED BINARY,
/*
NUMBER OF DEPENDENT CHILDREN                             */
2 DEP_CHILDREN FIXED BINARY,
/*
AGE RANGE OF THE DEPENDENT CHILDREN (YOUNGEST AND
OLDEST)                                                  */
2 AGE_RANGE(2) FIXED BINARY,
/*
WORK EXPERIENCES(MAXIMUM OF SIX EXPERIENCES):
TWO DIGIT CODE                                           */
2 WORK_EXP(6) FIXED BINARY,
/*
TRAVEL EXPERIENCES(MAXIMUM OF SIX EXPERIENCES):
TWO DIGIT CODE                                           */
2 TRAVEL_EXP(6) FIXED BINARY,
/*
CURRENT EMPLOYMENT                                       */
2 CUR_EMPLOYMENT,
  /*
  EMPLOYER'S NAME                                        */
  3 EMPLOYER CHARACTER(20),
  /*
  HOURS WORKED PER WEEK                                  */
  3 HRS_PER_WK FIXED BINARY,
/*
RECORD OF COLLEGES ATTENDED                              */
2 COL_ATT(4),
  /*
  NAME OF THE INSTITUTION                                */
  3 INSTITUTION CHAR(30),
  /*
  ATTENDANCE RECORD                                      */
  3 DATES,
    /*
    MATRICULATION DATE                                   */
    4 BEGIN CHARACTER(6),
    /*
    LAST REGISTRATION                                    */
    4 END CHARACTER(6),
/*
CURRENT GRADE POINT AVERAGE                              */
2 GPA FIXED BINARY,
/*
MYERS BRIGGS TYPE INDICATOR: TWO DIGIT CODE              */
2 MYERS_BRIGGS FIXED BINARY,
/*
DATE OF ENTRANCE INTO CEP                                */
2 CEP_ENTRANCE,
  /*
  YEAR OF ENTRANCE: LAST TWO DIGITS OF THE YEAR          */
  3 YR FIXED BINARY,
  /*
  QUARTER OF ENTRANCE: ONE DIGIT CODE                    */
  3 QTR FIXED BINARY,
  /*
  HOURS EXCEPTED TOWARD GRADUATION                       */
  3 HRS_COM FIXED DECIMAL(4,1),
/*
SPECIALIZATION: ONE DIGIT CODE                           */
2 SPECIALIZATION FIXED BINARY,
/*
TEAM NUMBER                                              */
2 TEAM FIXED BINARY,
/*
SEMINAR NUMBER                                           */
```

```
2 SEMINAR FIXED BINARY,
/*
EXPECTED DATE OF GRADUATION                        */
2 EXP_GRAD_DATE,
  /*
  EXPECTED YEAR OF GRADUATION: LAST TWO DIGITS
  OF THE YEAR                                      */
  3 YR FIXED BINARY,
  /*
  QUARTER OF GRADUATION: ONE DIGIT CODE            */
  3 QTR FIXED BINARY,
/*
GRADES RECEIVED FOR REQUIRED COURSES IN CEP        */
2 REQ_COURSE_GRADE(9) CHARACTER(1),
/*
REGISTRATION RECORD(MAXIMUM OF EIGHT REGISTRATION
RECORDS)                                           */
2 REGISTRATION(8),
  /*
  REGISTRATION DATE                                */
  3 DATE,
  /*
    YEAR OF REGISTRATION: LAST TWO DIGITS OF THE
    YEAR                                           */
    4 YR FIXED BINARY INITIAL((8)0),
    /*
    QUARTER OF REGISTRATION: ONE DIGIT CODE        */
    4 QTR FIXED BINARY INITIAL((8)0),
  /*
  COURSE REGISTRATION(MAXIMUM OF FIVE COURSES PER
  QUARTER)                                         */
  3 COURSES(5),
    /*
    COURSE NUMBER                                  */
    4 COURSE_NO CHARACTER(6) INITIAL((40)(1)' '),
    /*
    CREDIT HOURS                                   */
    4 HRS FIXED BINARY INITIAL((40)0),
    /*
    GRADE RECEIVED                                 */
    4 GRADE CHARACTER(1) INITIAL((40)(1)' '),
/*
LEARNING ACTIVITY RECORD CONSISTING OF 85
LEARNING ACTIVITIES, 3 BEYOND THE MINIMUMS AND 10
AREA SUMMARIES                                     */
2 LEARNING_A(98),
  /*
  FACULTY MEMBER WHO EVALUATED THIS ACTIVITY: TWO
  DIGIT FACULTY CODE                               */
  3 FACULTY FIXED BINARY INITIAL((118)0),
  /*
  DATE ACTIVITY WAS COMPLETED: SIX CHARACTER CODE
  (MONTH: TWO DIGIT CODE, DAY: TWO DIGITS,
  YEAR: LAST TWO DIGITS OF THE YEAR)               */
  3 DATE CHARACTER(6) INITIAL((118)(1)' '),
  /*
  FACULTY MEMBER'S COMMENTS ABOUT THIS ACTIVITY
  (MAXIMUM OF SIX COMMENTS): TWO DIGIT CODE        */
  3 COMMENT(6) FIXED BINARY INITIAL((708)0),
  /*
  ACTIVITY TITLE: THREE DIGIT CODE                 */
  3 ACTIVITY_TITLE FIXED BINARY INITIAL((118)0),
  /*
  GROUP STATUS: ONE BIT CODE                       */
  3 GROUP BIT(1) INITIAL((118)'0'B),
    /*
  SUBSTITUTE STATUS: ONE BIT CODE                  */
  3 SUBSTITUTE BIT(1) INITIAL((118)'0'B),
/*
FIELD EXPERIENCE RECORD(MAXIMUM OF SEVEN FIELD
EXPERIENCES)                                       */
2 FIELD(7),
  /*
  NAME OF THE SCHOOL: ONE DIGIT CODE               */
  3 SCHOOL FIXED BINARY INITIAL((7)0),
  /*
  GRADE LEVEL OF THE CLASSROOM: TWO DIGIT CODE     */
  3 GRADE FIXED BINARY INITIAL((7)0),
  /*
  LEVEL IN CEP: ONE DIGIT CODE                     */
  3 LEVEL FIXED BINARY INITIAL((7)0),
  /*
  DATE OF THIS FIELD EXPERIENCE                    */
  3 DATE,
                /*
                YEAR OF THIS FIELD EXPERIENCE: LAST TWO DIGITS
                OF THE YEAR                                         */
                4 YR FIXED BINARY INITIAL((7)0),
                /*
                QUARTER OF THIS FIELD EXPERIENCE: ONE DIGIT
                CODE                                                */
                4 QTR FIXED BINARY INITIAL((7)0),
              /*
              TYPE OF CLASSROOM: ONE DIGIT CODE                     */
              3 CLASSROOM FIXED BINARY INITIAL((7)0),
              /*
              NAME OF THE SUPERVISING TEACHER                       */
              3 TEACHER CHARACTER(14) INITIAL((7)(1)' '),
              /*
              ROLE(MAXIMUM OF SIX ROLES): TWO DIGIT CODE            */
              3 F_ROLE(6) FIXED BINARY INITIAL((42)0),
              /*
              STRENGTHS(MAXIMUM OF SIX STRENGTHS): TWO DIGIT
              CODE                                                  */
              3 STRENGTHS(6) FIXED BINARY INITIAL((42)0),
              /*
              WEAKNESSES(MAXIMUM OF SIX WEAKNESSES): TWO DIGIT
              CODE                                                  */
              3 WEAKNESSES(6) FIXED BINARY INITIAL((42)0),
            /*
            COMMUNITY EXPERIENCE                                    */
            2 COMMUNITY,
              /*
              AGENCY NAME                                           */
              3 AGENCY CHARACTER(20) INITIAL(' '),
              /*
              NAME OF THE AGENCY REPRESENTATIVE                     */
              3 AGENCY_REP CHARACTER(14) INITIAL(' '),
              /*
              LEVEL IN CEP: ONE DIGIT CODE                          */
              3 LEVEL FIXED BINARY INITIAL(0),
              /*
              DATE OF THE COMMUNITY EXPERIENCE                      */
              3 DATE,
                /*
                YEAR OF THE COMMUNITY EXPERIENCE: LAST TWO
                DIGITS OF THE YEAR                                  */
                4 YR FIXED BINARY INITIAL(0),
                /*
                QUARTER OF THE COMMUNITY EXPERIENCE: ONE DIGIT
                CODE                                                */
                4 QTR FIXED BINARY INITIAL(0),
              /*
              ROLE(MAXIMUM OF FOUR ROLES): TWO DIGIT CODE           */
              3 C_ROLE(4) FIXED BINARY INITIAL((4)0),
              /*
              AGENCY REPRESENTATIVE'S COMMENTS(MAXIMUM OF FOUR
              COMMENTS): TWO DIGIT CODE                             */
              3 C_COMMENT(4) FIXED BINARY INITIAL((4)0);

TWC:    DECLARE
        /*
          DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
          CODES RECORDED IN PERSONAL FOR LEARNING ACTIVITY
          TITLES.                                                   */
          ACTIVITY_TITLE_DICT(0:200) CHARACTER(46) INITIAL
            (' ',
            'ORIENTATION',
            'FINAL EVALUATION',
            'NEED AND ROLE OF ART',
            'ART GROWTH AND DEVELOPMENT',
            'ART CURRICULUM PLANNING',
            'STUDIO ACTIVITIES',
            (24)(1)' ',
            'BASES FOR CURRICULUM PAPER',
            'MEDIA SEARCH',
            'SOCIAL CLASS INTERVIEWS',
            'STUDY OF ONE CHILD',
            'REPORTING PUPIL PROGRESS',
            'TEACHING LIFE PROCESSES',
            'CREATIVE THINKING MODULE',
            'ORGANIZING LEARNING EXPERIENCES',
            'ROLE OF TEACHER',
            'GROUPING FOR INSTRUCTION',
            'PLANS FOR TEACHING-LEARNING SETTING',
            'DEVELOPING INSTRUCTIONAL STRATEGIES',
            (8)(1)' ',
            'LISTENING AND SPEAKING',
'NATURE OF LANGUAGE',
'VISUAL LITERACY',
'HANDWRITING AND SPELLING',
'COMPOSITION AND GRAMMAR',
'CHILDRENS LANGUAGE',
'SHARING LEARNING CENTERS',
'CREATIVITY',
(12)(1)' ',
'BEGINNING NUMBERS',
'ADDITION - SUBTRACTION',
'MULTIPLICATION - DIVISION',
'FRACTIONS',
'DECIMALS AND PERCENT',
'MEASUREMENT AND GEOMETRY',
'NUMERATION SYSTEMS AND BASES',
(13)(1)' ',
'ANCILLARY INSTRUMENTS',
'TEACHING',
'RESOURCE COLLECTION',
'INTERRELATED TEACHING UNIT',
'CARL ORFF APPROACH',
'CREATIVE COMPOSITIONAL APPROACH',
'ZOLTON KODOLY APPROACH',
'FUNDAMENTAL, CREATIVE AND DIRECTED MOVEMENT',
'LEARNING CENTERS AND IPBI IN MUSIC',
'PHILOSOPHY PAPER',
'MAKING AND PLAYING THE DULCIMER',
'COMPOSITIONAL APPROACH',
'MUSICAL CONCEPTS FOR EARLY CHILDHOOD',
'MODELS FOR TEACHING',
'MUSIC AND OPEN EDUCATION',
'ROCK,JAZZ, GOSPEL:" LET THEM TEACH',
'ORFF GUIDED PROJECT',
(2)(1)' ',
'ACTIVITY 1',
'ACTIVITY 2',
'ACTIVITY 3',
'ACTIVITY 4',
'ACTIVITY 5',
'ACTIVITY 6',
'ACTIVITY 7',
'ACTIVITY 8',
'ACTIVITY 9',
'ACTIVITY 10',
'APPLICATION OF PRINCIPLES OF HUMAN DEVELOPMENT',
'COMPLETED BY TAKING EDF345',
(8)(1)' ',
'TERMINOLOGY AND WORD ATTACK SKILLS',
'ASSESS PUPILS READING LEVELS',
'SELECTION OF APPROPRIATE MATERIALS',
'DETERMINE APPROPRIATE SKILLS AND TEACH THEM',
'CLASSROOM ORGANIZATION AND RECORD KEEPING',
'PREPARATION OF MATERIAL FILE',
'READING READINESS',
'METHODS OF TEACHING READING',
(12)(1)' ',
'LABORATORY 1',
'LABORATORY 2',
'LABORATORY 3',
'LABORATORY 4',
'GRAPHING VARIABLES',
'MEASURING',
'DESIGNING AND RUNNING EXPERIMENTS',
'ECOLOGICAL VARIABLES',
'CONTROLLING VARIABLES',
'WAIT TIME AS A FACTOR IN INQUIRY',
'FATE CONTROL',
'VERBAL SANCTIONS',
'INQUIRY',
'CLASSROOM CONTROL',
'MANAGEMENT',
'EXEMPT BY PREVIOUS REQUIREMENTS',
(4)(1)' ',
'KEEPING UP WITH RESEARCH',
'SYSTEMATIC CLASSROOM OBSERVATION',
'MAPPING AND AREA SOCIAL SYSTEMS',
'COMPLETED BY TAKING EDF320',
(5)(1)' ',
'SOCIAL SCIENCE DISCIPLINES',
'MEDIA AND ACTIVITIES',
'SKILLS OF SOCIAL STUDIES',
'PROGRAM PLANNING AND EVALUATION',
'PERSONAL-SOCIAL AND INTERGROUP PROBLEM SOLVING',
'GROUP STUDIES AND PROBLEM SOLVING',
   (4)(1)' ',
   'ART',
   'CURRICULUM',
   'HUMAN GROWTH AND DEVELOPMENT',
   'LANGUAGE ARTS',
   'MATHEMATICS',
   'MUSIC',
   'READING',
   'SCIENCE',
   'SOCIAL FOUNDATIONS',
   'SOCIAL STUDIES',
   (1)(1)' '),
/*
DICTIONARY OF THE SUBSTANTIVE AREA NAMES ASSOCIATED
WITH THE CODES USED IN PERSONAL.                              */
AREA_DICT(11) CHARACTER(21) INITIAL('ART','CURRICULUM'
   ,'HUMAN GROWTH AND DEV.','LANGUAGE ARTS','MATH',
   'MUSIC','READING','SCIENCE','SOCIAL FOUNDATIONS',
   'SOCIAL STUDIES','BEYOND MINIMUM'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR COMMENTS ABOUT
COMMUNITY EXPERIENCES.                                        */
C_COMMENT_DICT(0:10) CHARACTER(26) INITIAL
   (' ',
   'COMMON SENSE',
   'COOPERATIVE',
   'CREATIVE THINKING',
   'ENTHUSIASTIC',
   'INDUSTRIOUS',
   'INITIATIVE',
   'RELATES WELL WITH CHILDREN',
   'WORKS WELL WITH CHILDREN',
   'WORKS WELL WITH COWORKERS',
   'WORKS WELL WITH SUPERVISOR'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR THE ROLES PLAYED
DURING A COMMUNITY EXPERIENCE.                                */
C_ROLE_DICT(0:15) CHARACTER(42) INITIAL
   (' ',
   'ANSWERED PHONE',
   'ASSISTED IN SPECIAL CLASSES',
   'COLLECTED SAMPLES OF DISPLAY USAGE',
   'CONDUCTED TOURS',
   'GENERAL NURSERY ACTIVITIES',
   'OBSERVED',
   'ORGANIZED GAMES',
   'PART TIME LIBRARIAN',
   'PLAYED WITH CHILDREN',
   'VISITED PATIENTS',
   'WORKED WITH ARTS AND CRAFTS',
   'WORKED WITH FREE PLAY ACTIVITIES',
   'WORKED WITH MUSIC AND READINESS ACTIVITIES',
   'WROTE ANCEDOTAL SITUIATIONS',
   (1)(1)' '),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR THE TYPE OF CLASS-
ROOM.                                                         */
CLASSROOM_DICT(0:4) CHARACTER(12) INITIAL
   (' ','BASIC SKILLS','FREE','OPEN','STANDARD'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR COMMENTS ABOUT
LEARNING ACTIVITIES.                                          */
COMMENT_DICT(0:60) CHARACTER(40) INITIAL
   (' ',
   'ACCURATE',
   'ADAPTABLE',
   'BRIEF',
   'CAREFUL IDEAS',
   'COMMAND OF SUBJECT',
   'COMPETENT',
   'COMPLETE',
   'CREATIVE',
   'CRITICAL INDEPENDENT THINKER',
   'DESCRIPTIVE',
   'ENERGETIC',
   'ENTHUSIASTIC',
   'EXCELLENT',
   'EXCELLENT IDEAS',
   'FAIR',
   'FLEXIBLE',
  'GOOD',
  'GOOD CLASSROOM APPLICATIONS',
  'GOOD CRITERIA',
  'IMAGINATIVE',
  'INTERESTING',
  'LOGICAL',
  'MOTIVATED',
  'NEED MORE EVIDENCE TO SUPPORT CONCLUSION',
  'NEEDS MORE DOCUMENTATION',
  'NEEDS TO BE MORE CONCISE',
  'NOT ENOUGH SOURCES',
  'ORIGINAL IDEAS',
  'PRECISE',
  'POOR',
  'REASONED INFERENCE',
  'RELIABLE',
  'RELATES WELL TO PEOPLE',
  'RESOURCEFUL',
  'SATISFACTORY',
  'SEQUITER',
  'SCIENTIFIC',
  'SCHOLARLY',
  'STEADY PURSUIT OF TASK',
  'STIMULATING',
  'SYSTEMATIC',
  'THOUGHTFULLY APPLIED',
  'TOO GENERAL',
  'WELL CHOSEN READINGS',
  'WELL DISCUSSED',
  'WELL DONE',
  'WELL RESEARCHED',
  'WELL STATED',
  'WORK COMPETENT IF ATTENTIVE',
  'WORK SHOWS HUMOR',
  'WRITING DIFFICULT TO READ',
  ' ',
  'ABILITY TO DO THIS',
  'INTEREST IN WORKING WITH PUPILS',
  'NO PROBLEMS',
  'UNDERSTOOD ACTIVITY',
  'PASSED',
  'WORKS HARD',
  'INTERESTED',
  (1)(1)' '),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR THE ROLES PLAYED
DURING A FIELD EXPERIENCE.                              */
F_ROLE_DICT(0:55) CHARACTER(44) INITIAL
  (' ',
  'ADMINISTERED SOCIO-GRAM',
  'ADMINISTERED STANDARDIZED TEST',
  'ADMINISTERED TESTS',
  'ARRANGED CLASSROOM ENVIORNMENT',
  'ARRANGED SCHOOL VISIT FROM A RESOURCE PERSON',
  'ATTENDED SCHOOL FUNCTION',
  'DEVELOPED LEARNING CENTERS',
  'DIRECTED PRODUCTION OF A PLAY OR PROGRAM',
  'DIRECTED SHARING ACTIVITIES',
  'ENRICHMENT ACTIVITIES',
  'EVALUATED PAPERS',
  'EVALUATED STUDENT PROGRESS',
  'GEN. ACTIVITIES AND PROCEDURES',
  'INDIVIDUAL INSTRUCTION',
  'LARGE GROUP INSTRUCTION',
  'LEAD CLASSROOM DISCUSSION',
  'MADE HOME VISIT',
  'MADE MATERIALS AND/OR DISPLAYS',
  'OBSERVED CLASS',
  'OBSERVED PSYCH SERVICES',
  'OBSERVED TEAM TEACHING',
  'PARENT TEACHER CONFERENCE',
  'PLANNED ACTIVITIES AROUND UNITS',
  'PLANNED AND SUPERVISED ACTIVITIES',
  'PLANNED GAMES AND ACTIVITIES',
  'PLANNED GROUP INSTRUCTION',
  'PLANNED WORK TIME',
  'PLAYED GAMES WITH CHILDREN',
  'PREPARED INDEPENDENT WORK',
  'PREPARED LESSONS FOR CLASS',
  'READ STORIES TO CLASS',
  'SMALL GROUP INSTRUCTION',
  'SOME TEAM TEACHING',
  'TAUGHT ART',
     'TAUGHT LANGUAGE ARTS',
     'TAUGHT MATH',
     'TAUGHT MUSIC',
     'TAUGHT P.E.',
     'TAUGHT SCIENCE',
     'TAUGHT SOCIAL STUDIES',
     'TAUGHT READING',
     'TEAM PLANNING',
     'SMALL GROUP INSTRUCTION',
     'USED AUDIO-VISUAL EQUIPMENT',
     'WORKED IN ALL AREAS',
     'WORKED WITH EMR CHILDREN',
     'WORKED WITH FOREIGN STUDENTS',
     'WORKED WITH GROUPS',
     'WORKED WITH LEARNING CENTERS',
     'USED DIFFERENT TEACHING TECHNIQUES',
     'INDIVIDUAL CLASSROOM CONFERENCES',
     (4)(1)' '),
  /*
  DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
  CODES RECORDED IN PERSONAL FOR GPA.                          */
  GPA_DICT(0:6) CHARACTER(11) INITIAL
     (' ',
     '1.5 OR LESS',
     '1.6 - 2.0',
     '2.1 - 2.5',
     '2.6 - 3.0',
     '3.1 - 3.5',
     '3.6 - 4.0'),
  /*
  DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
  CODES RECORDED IN PERSONAL FOR THE GRADE LEVEL OF A
  CLASSROOM.                                                   */
  GRADE_DICT(17) CHARACTER(18) INITIAL('FIRST','SECOND',
     'THIRD','FOURTH','FIFTH','SIXTH','SEVENTH','EIGHTH',
     'PRE-KINDERGARTEN','KINDERGARTEN',
     'MULTI-AGE PRIMARY','MULTI-AGE UPPER ELE',
     'MIDDLE SCHOOL','SPECIAL EDUCATION','MUSIC',
     'READING','PHYSICAL EDUCATION'),
  /*
  DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
  CODES RECORDED IN PERSONAL FOR GROUP.                        */
  GP(2) CHARACTER(10) INITIAL('GROUP','INDIVIDUAL'),
  /*
  DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
  CODES RECORDED IN PERSONAL FOR LEVEL IN CEP.                 */
  LEVEL_DICT(0:4) CHARACTER(9) INITIAL(' ','INITIATE',
     'ASSISTANT','ASSOCIATE','INTENSIVE'),
  /*
  DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
  CODES RECORDED IN PERSONAL FOR MARITAL STATUS.               */
  MARITAL_STAT_DICT(0:4) CHARACTER(8) INITIAL
     (' ',
     'SINGLE',
     'MARRIED',
     'DIVORCED',
     'WIDOWED'),
  /*
  DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
  CODES RECORDED IN PERSONAL FOR THE MYERS BRIGGS
  TYPE.                                                        */
  MYERS_BRIGGS_DICT(0:16) CHARACTER(4) INITIAL('','ISTJ'
     ,'ISTP','ESTP','ESTJ','ISFJ','ISFP','ESFP','ESFJ',
     'INFJ','INFP','ENFP','ENFJ','INTJ','INTP','ENTP',
     'ENTJ'),
  /*
  DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
  CODES RECORDED IN PERSONAL FOR THE QUARTER.                  */
  QUARTER_DICT(0:4) CHARACTER(6) INITIAL
     (' ',
     'FALL',
     'WINTER',
     'SPRING',
     'SUMMER'),
  /*
  DICTIONARY OF REQUIRED COURSE NAMES ASSOCIATED WITH
  THE CODES RECORDED IN PERSONAL.                              */
  REQ_COURSE_DICT(9) CHARACTER(17) INITIAL('EH491',
     'MSC260','MSC300','MS310/311','PHR361','PHR373',
     'SCH201','MEDIA EQUIPMENT','MEDIA PRODUCTIONS'),
  /*
  DICTIONARY OF THE NAMES ASSOCIATED WITH THE REQUEST
  CODES.                                                       */
  REQUEST_DICT(8) CHARACTER(13) INITIAL('MYERS BRIGGS',
   'REQ. COURSES','REGISTRATION','L.A.','L.A. SUMMARY',
   'FIELD','FIELD SUMMARY','COMMUNITY'),
/*
DICTIONARY OF SCHOOL NAMES ASSOCIATED WITH THE CODES
RECORDED IN PERSONAL)                                      */
SCHOOL_DICT(7) CHARACTER(13) INITIAL('J.J. FINLEY',
   'HAWTHORNE','SIDNEY LANIER','LITTLEWOOD','SHELL',
   'TEWILLIGER','WESTWOOD'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR THE AREA OF
SPECIALIZATION.                                            */
SPECIALIZATION_DICT(2) CHARACTER(15) INITIAL
   ('ELEMENTARY','EARLY CHILDHOOD'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR THE STRENGTHS DISPLAYED
DURING A FIELD EXPERIENCE.                                 */
STRENGTHS_DICT(0:75) CHARACTER(42) INITIAL
   (' ',
   'ADAPTS MATERIALS TO ACHIEVEMENT LEVELS',
   'AWARE OF INDIVIDUAL INTERESTS',
   'AWARE OF INDIVIDUAL NEEDS',
   'AWARE OF INDIVIDUAL PROBLEMS',
   'CAPABLE',
   'CHILDREN LIKE HER(HIM)',
   'CONCERNED WITH CHILDS WELFARE',
   'CONFIDENT',
   'CONSTRUCTS GOOD EVALUATION INSTRUMENTS',
   'COOPERATIVE',
   'CREATIVE',
   'DEPENDABLE',
   'DESIRE TO HELP CHILDREN',
   'DESIRE TO LEARN',
   'EFFECTIVE CLASSROOM CONTROL',
   'EMPHASIZES INDIVIDUAL NEEDS',
   'EVALUATES ACCURATELY',
   'EXCELLENT PLANNER',
   'EXCELLENT RAPPORT',
   'EXCELLENT RELATION WITH STAFF',
   'EXCELLENT RELATION WITH PARENTS',
   'EXCEPTIONALLY CAPABLE IN ART',
   'EXCEPTIONALLY CAPABLE IN LANG.A.',
   'EXCEPTIONALLY CAPABLE IN MATH',
   'EXCEPTIONALLY CAPABLE IN MUSIC',
   'EXCEPTIONALLY CAPABLE IN READING',
   'EXCEPTIONALLY CAPABLE IN SCIENCE',
   'EXCEPTIONALLY CAPABLE IN S.S.',
   'FLEXIBLE',
   'GAINS STUDENTS TRUST',
   'GOES BEYOND WHAT IS EXPECTED',
   'GOOD RAPPORT',
   'GOOD VOICE CONTROL',
   'GOOD WITH CHILDREN',
   'GROWTH IN DEPENDIBILITY',
   'HELPS STUDENTS FEEL SUCCESSFUL',
   'INNOVATIVE',
   'INTEREST IN CHILDRENS WORK',
   'INITIATES CLASS INVOLVEMENT',
   'KNOWLEDGEABLE',
   'LARGE GROUP WORK',
   'LIKES THIS AGE GROUP',
   'LIKES WORKING WITH CHILDREN',
   'MOTIVATES AND INTERESTS STUDENTS',
   'PATIENT',
   'PERSERVERENCE',
   'PLANS EFFECTIVELY',
   'POSITIVE ATTITUDE',
   'PROFESSIONAL PROMISE',
   'PROMPT',
   'RESPONDS WELL TO CRITICISM',
   'SEES WHAT NEEDS TO BE DONE',
   'SENSE OF RESPONSIBILITY',
   'SENSITIVE TO STUDENTS FEELINGS',
   'SHOWS INITIATIVE',
   'SMALL GROUP WORK',
   'SYMPATHETIC',
   'TEACHES FROM CHILDRENS INTERESTS',
   'TREATS CHILDREN FAIRLY',
   'UNDERSTANDS CHILDREN',
   'USES INDIVIDUAL DIFFERENCES',
   'USES OUTSIDE RESOURCES WELL',
   'USES PEER TEACHING',
   'VERY RESPONSIBLE',
   'WANTS CHILDREN TO UNDERSTAND CONCEPTS',
   'WELL ORGANIZED',
   'WELL PREPARED',
   'WORK WITH LEARNING PROBLEMS',
   'WORK IN ALL PHASES OF CLASSROOM ACTIVITIES',
   'EFFICIENT',
   'ENTHUSIASTIC',
   'FRIENDLY',
   'GOOD COMMAND OF ENGLISH',
   'GOOD IDEAS',
   'POISED'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR SUBSTITUTE.               */
SOE(2) CHARACTER(10) INITIAL('SUBSTITUTE','REGULAR'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR TRAVEL EXPERIENCE.        */
TRAVEL_EXP_DICT(0:19) CHARACTER(22) INITIAL
  (' ',
   'AFRICA',
   'ASIA',
   'BERMUDA AND BAHAMAS',
   'CANADA',
   'CARIBBEAN',
   'CENTRAL AMERICA',
   'EUROPE',
   'MEXICO',
   'MIDDLE EAST',
   'PACIFIC ISLANDS',
   'U.S.A.- ALL OVER',
   'FAR WESTERN STATES',
   'MIDDLE ATLANTIC STATES',
   'MIDWESTERN STATES',
   'NEW ENGLAND STATES',
   'PLAINS STATES',
   'ROCKY MOUNTAIN STATES',
   'SOUTHEASTERN STATES',
   'SOUTHWESTERN STATES'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR WORK EXPERIENCE.          */
WORK_EXP_DICT(0:18) CHARACTER(25) INITIAL
  (' ',
   'BABYSITTER',
   'CAMP COUNSELOR',
   'CLERK',
   'CLUB WORK',
   'DAY CARE CENTER',
   'LABORATORY TECHNICIAN',
   'LITTLE THEATRE',
   'NURSERY SCHOOL',
   'RECEPTIONIST',
   'RECREATION CENTER',
   'SALESPERSON',
   'SECRETARY',
   'STUDENT ASSISTANT',
   'SUBSTITUTE',
   'TEACHERS AID',
   'TOUR GUIDE',
   'TUTOR',
   'OTHER WORK CLASSIFICATION'),
/*
DICTIONARY OF THE ITEM NAMES ASSOCIATED WITH THE
CODES RECORDED IN PERSONAL FOR THE NEEDS DISPLAYED
DURING A FIELD EXPERIENCE.                               */
WEAKNESSES_DICT(0:70) CHARACTER(35) INITIAL
  (' ',
   'ART EXPERIENCE',
   'BETTER RAPPORT WITH STUDENTS',
   'BETTER RAPPORT WITH STAFF',
   'BETTER RAPPORT WITH PARENTS',
   'BETTER UNDERSTANDING OF CHILDREN',
   'BETTER VOICE CONTROL',
   'CHALKBOARD EXPERIENCE',
   'CONCERN FOR CHILDRENS WELFARE',
   'DISCIPLINE EXPERIENCE',
   'EVALUATION EXPERIENCE',
   'EXPERIENCE DIRECTING LESSONS',
   'EXPOSURE TO LOWER GRADES',
   'EXPOSURE TO MORE GRADES',
   'EXPOSURE TO UPPER GRADES',
   'GROUP MANAGEMENT EXPERIENCE',
   'LANGUAGE ARTS EXPERIENCE',
      'LARGE GROUP EXPERIENCE',
      'LOWER INCOME EXPERIENCE',
      'MATH EXPERIENCE',
      'MORE ENTHUSIASM',
      'MORE PERSERVERENCE',
      'MORE TEACHING TECHNIQUES',
      'MUSIC EXPERIENCE',
      'P.E. EXPERIENCE',
      'POSITIVE ATTITUDE',
      'READING EXPERIENCE',
      'SCIENCE EXPERIENCE',
      'SMALL GROUP EXPERIENCE',
      'SOCIAL STUDIES EXPERIENCE',
      'TEACHING EXPERIENCE',
      'TESTING EXPERIENCE',
      'TO ADAPT BETTER',
      'TO BE AWARE OF INDIVIDUAL INTERESTS',
      'TO BE AWARE OF INDIVIDUAL PROBLEMS',
      'TO BE CONFIDENT',
      'TO BE CREATIVE',
      'TO BE DEPENDABLE',
      'TO BE INNOVATIVE',
      'TO BE LESS SENSITIVE',
      'TO BE ORGANIZED',
      'TO BE MORE SENSITIVE',
      'TO BE PATIENT',
      'TO BE PREPARED',
      'TO BE PUNCTUAL',
      'TO BE RESPONSIBLE',
      'TO BE RELAXED',
      'TO BE WILLING TO HELP CHILDREN',
      'TO BE WILLING TO HELP COWORKERS',
      'TO COMPLETE PROJECTS',
      'TO CONSTRUCT EVALUATION INSTRUMENTS',
      'TO CONTROL LARGE GROUPS',
      'TO EMPHASIZE INDIVIDUAL NEEDS',
      'TO HELP STUDENTS FEEL SUCCESSFUL',
      'TO INITIATE CLASS INVOLVEMENT',
      'TO PLAN BETTER',
      'TO PREPARE UNITS',
      'TO RECOGNIZE INDIVIDUAL NEEDS',
      'TO RELATE BETTER',
      'TO RESPOND BETTER TO CRITICISM',
      'TO SHOW INITIATIVE',
      'TO TEACH FROM CHILDRENS INTEREST',
      'TO TREAT STUDENTS FAIRLY',
      'TO USE OUTSIDE RESOURCES BETTER',
      'TO USE PEER TEACHING',
      'WORK IN CREATIVE AREAS',
      'WORK WITH LEARNING PROBLEMS',
      'WORK IN ALL AREAS',
      'WORK WITH DIFFERENT MATERIALS',
      (2)(1)' ');
 DECLARE
   /*
   DICTIONARY OF THE FACULTY MEMBER'S NAMES ASSOCIATED
   WITH THE CODES RECORDED IN PERSONAL.                    */
   FACULTY(26) CHARACTER(14) INITIAL('BERNARD','CURRAN',
      'DETURE','FEY','GOLDMAN','HILLIARD','JESTER',
      'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
      'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
      'TISON','WILMOT','KRALL','CREEDON','LEVINE',
      'HEDGES','GRIMM','DORMAN'),
   /*
   FACULTY NUMBERS OF THE SEMINAR LEADERS BY TEAM.         */
   FACULTY_NO(MIN:MAX,4) FIXED BINARY,
   /*
   AN ENTRY POINT, FOR INSTANCE ENTRY(I), INDICATES THE
   NUMBER OF THE POSITION PRECEEDING THE BEGINNING OF
   THIS SUBSTANTIVE AREA IN LEARNING_A.                    */
   ENTRY(12) FIXED BINARY INITIAL(0,6,19,31,41,50,56,66,
      82,87,95,99),
   CUR_DATE CHARACTER(8),
   FACULTY# FIXED BINARY,
   OLD FILE RECORD SEQUENTIAL,
   FLAG BIT(1) INITIAL('0'B),
   SEX_NAME CHARACTER(7),
   SS_NO CHARACTER(9);
 /*

 INITIALIZE FACULTY.

                                                           */
 FACULTY_NO = 0;
 OPEN FILE(SYSPRINT) LINESIZE(130);
```

```
         /*

         ON ENCOUNTERING AN END OF FILE ON OLD, REWIND THE TAPE.
         IF FLAG IS SET TO '1'B, READ THE NEXT SOCIAL SECURITY
         NUMBER.  IF SS_NO IS AN '*', RESET FLAG TO '0'B AND
         OBTAIN A NEW REQUEST, OTHERWISE CONTINUE PROCESSING THE
         SOCIAL SECURITY NUMBERS.
                                                                 */
         ON ENDFILE(OLD) BEGIN;
           CLOSE FILE(OLD);
           OPEN FILE(OLD);
           IF FLAG = '1'B THEN DO;
             GET FILE(SYSIN) EDIT(SS_NO)(COL(1),A(9));
             IF SS_NO = '*' THEN DO;
               FLAG = '0'B;
               GO TO NEXT;
               END;
               ELSE GO TO LOOP;
             END;
           GO TO NEXT;
           END;
         /*

         ON ENCOUNTERING A SUBSCRIPTRANGE ERROR, PRINT AN ERROR
         MESSAGE AND TRANSFER TO PROCESS THE NEXT RECORD FOR
         THIS PARTICULAR REQUEST.                                */
         ON SUBSCRIPTRANGE BEGIN;
           PUT FILE(SYSPRINT) EDIT(SOC_SEC_NO,'SUBSCRIPT OUTSIDE'
             ,' LEGAL BOUNDS')(A,COL(32),2 A);
           IF REQUEST_TYPE = 1 THEN GO TO GROUPS;
             ELSE GO TO SINGLE;
           END;
         /*

         ON ENCOUNTERING THE END OF THE REQUESTS TRANSFER TO
         FINISH.
                                                                 */
         ON ENDFILE(SYSIN) GO TO FINISH;
         /*

         ON ENCOUNTERING THE END OF A PAGE OF PRINT, PRINT THE
         HEADER INFORMATION.
                                                                 */
         ON ENDPAGE(SYSPRINT) BEGIN;
           PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
           PUT FILE(SYSPRINT) EDIT('ALL','OUTPUT ',
             'DESCRIPTION: LISTING OF ALL DATA ON STUDENTS AS ',
             'REQUESTED BY FACULTY MEMBERS','REQUEST FOR ',
             FACULTY(FACULTY#))(SKIP,A,SKIP,3 A,SKIP,2 A);
           PUT FILE(SYSPRINT) EDIT('S.S. NUMBER AND NAME','ITEM',
             'CONTENTS','______________________','____','________')
             (SKIP(6),2 (COL(1),A,X(12),A,X(11),A,SKIP(0)));
           PUT FILE(SYSPRINT) SKIP(2);
           END;
         /*

         READ THE CURRENT DATE.
                                                                 */
         GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
         /*

         READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
         TEAM READ THE TEAM NUMBER, THE NUMBER OF SEMINARS TO BE
         PROCESSED, AND THE SEMINAR NUMBER AND FACULTY NUMBER
         OF EACH SEMINAR TO BE PROCESSED.
                                                                 */
         GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
         DO I = 1 TO NO_CARDS;
           GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
             TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
             X(4)),4 (F(1),X(4),F(2),X(3)));
           END;
         /*

         READ THE INITIAL DATA FOR THE NEXT FACULTY REQUEST.
                                                                 */
 NEXT:   GET FILE(SYSIN) EDIT(FACULTY#,REQUEST_TYPE,TEAM#,
           SEMINAR#)(COL(1),F(2),X(3),3 (F(1),X(4)));
         /*

         IF THE REQUEST TYPE IS ONE, PROCESSING IS TO BE DONE BY
         SEMINARS.
                                                                 */
         IF REQUEST_TYPE = 1 THEN DO;
```

```
          /*

          READ A RECORD FROM THE MAIN FILE.
                                                                */
GROUPS:   READ FILE(OLD) INTO(PERSONAL);
          /*

          CHECK IF THIS STUDENT IS IN THE TEAM AND SEMINAR
          CURRENTLY BEING PRINTED.
                                                                */
          IF PERSONAL.TEAM = TEAM# & PERSONAL.SEMINAR = SEMINAR#
            THEN DO;
            /*

            PRINT THE HEADER INFORMATION.
                                                                */
            PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
            PUT FILE(SYSPRINT) EDIT('ALL','OUTPUT ',
              'DESCRIPTION: LISTING OF ALL DATA ON STUDENTS AS '
              ,'REQUESTED BY FACULTY MEMBERS','REQUEST FOR ',
              FACULTY(FACULTY#))(SKIP,A,SKIP,3 A,SKIP,2 A);
            /*

            PRINT THE STUDENT'S SOCIAL SECURITY NUMBER AND NAME.
                                                                */
            PUT FILE(SYSPRINT) EDIT('S.S. NUMBER AND NAME',
              'ITEM','CONTENTS','______________________','____',
              '__________')(SKIP(6),2 (COL(1),A,X(12),A,X(11),A,
              SKIP(0)));
            PUT FILE(SYSPRINT) EDIT(SOC_SEC_NO,SUBSTR(
              PERSONAL.LAST,1,INDEX(PERSONAL.LAST,' ')-1) || ','
              || PERSONAL.FIRST)(SKIP(2),A,COL(6),A);
            /*

          CALL THE PROCEDURE PRINT TO PRINT INFORMATION ABOUT
          THIS STUDENT.
                                                                */
            CALL PRINT;
            END;
          /*

          TRANSFER TO GROUPS TO PROCESS A NEW STUDENT RECORD.
                                                                */
          GO TO GROUPS;
          END;
          /*

          IF THE REQUEST TYPE IF ZERO, PROCESSING IS TO BE DONE
          INDIVIDUAL SOCIAL SECURITY NUMBERS.
                                                                */
                            ELSE DO;
          FLAG = '1'B;
          /*

          READ THE SOCIAL SECURITY NUMBER OF THE NEXT RECORD TO
          BE PRINTED.
                                                                */
SINGLE:   GET FILE(SYSIN) EDIT(SS_NO)(COL(1),A(9));
          /*

          IF THE SOCIAL SECURITY NUMBER IS AN '*', WHICH
          INDICATES THE END OF THE INDIVIDUAL REQUESTS, REWIND
          THE TAPE AND TRANSFER TO READ THE NEXT REQUEST.
                                                                */
          IF SS_NO = '*' THEN DO;
            CLOSE FILE(OLD);
            OPEN FILE(OLD);
            GO TO NEXT;
            END;
          /*

          PRINT THE HEADER INFORMATION;
                                                                */
          PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
          PUT FILE(SYSPRINT) EDIT('ALL','OUTPUT ',
            'DESCRIPTION: LISTING OF ALL DATA ON STUDENTS AS ',
            'REQUESTED BY FACULTY MEMBERS','REQUEST FOR ',
           FACULTY(FACULTY#))(SKIP,A,SKIP,3 A,SKIP,2 A);
          PUT FILE(SYSPRINT) EDIT('S.S. NUMBER AND NAME','ITEM',
            'CONTENTS','______________________','____','__________')
            (SKIP(6),2 (COL(1),A,X(12),A,X(11),A,SKIP(0)));
          /*

          READ A RECORD FROM THE MAIN FILE.
```

```
                                                              */
LOOP:     READ FILE(OLD) INTO(PERSONAL);
          /*

          IF THE RECORD TO BE PRINTED HAS NOT YET BEEN FOUND IN
          THE MAIN FILE, GET A NEW RECORD.
                                                              */
          IF PERSONAL.SOC_SEC_NO < SS_NO THEN GO TO LOOP;
          /*

          IF THE RECORD TO BE PRINTED DOES NOT EXIST IN THE MAIN
          FILE, PRINT THE SOCIAL SECURITY NUMBER WITH THE
          MESSAGE '*MISSING'.  READ THE NEXT SOCIAL SECURITY
          NUMBER.  IF IT IS AN '*', REWIND THE TAPE AND OBTAIN A
          NEW REQUEST.
                                                              */
          IF PERSONAL.SOC_SEC_NO > SS_NO THEN DO;
            PUT FILE(SYSPRINT) EDIT(SS_NO,'*MISSING')(SKIP(4),
              A,X(20),A);
            GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
            IF SS_NO = '*' THEN DO;
              CLOSE FILE(OLD);
              OPEN FILE(OLD);
              GO TO NEXT;
              END;
            GO TO LOOP;
            END;
          /*

          PRINT THE STUDENT'S SOCIAL SECURITY NUMBER AND NAME.
                                                              */
          PUT FILE(SYSPRINT) EDIT(SOC_SEC_NO,SUBSTR(
            PERSONAL.LAST,1,INDEX(PERSONAL.LAST,' ')-1) || ','
            || PERSONAL.FIRST)(SKIP(2),A,COL(6),A);
          /*

          CALL THE PROCEDURE PRINT TO PRINT INFORMATION ABOUT
          THIS STUDENT.
                                                              */
          CALL PRINT;
          /*

          TRANSFER TO SINGLE TO PROCESS A NEW STUDENT RECORD.
                                                              */
          GO TO SINGLE;
          END;
        /*

        THE PROCEDURE PRINT PRINTS ALL INFORMATION CONTAINED IN
        A STUDENT'S RECORD.
                                                              */
PRINT:  PROCEDURE;
        /*

        PRINT THE STUDENT'S PERMANENT ADDRESS.
                                                              */
        PUT FILE(SYSPRINT) EDIT('HOME ADDR.')(COL(32),A);
        PERM_LENGTH = INDEX(PERM_ADDR.CITY,'  ');
        IF PERM_LENGTH = 0 THEN PERM_LENGTH = 20;
        PUT FILE(SYSPRINT) EDIT(PERM_ADDR.STREET,
          SUBSTR(PERM_ADDR.CITY,1,PERM_LENGTH
          -1),', ',PERM_ADDR.STATE,PERM_ADDR.ZIP)(COL(46),A,
          COL(46),3 A,X(1),A);
        /*

        PRINT THE STUDENT'S PERMANENT PHONE NUMBER.
                                                              */
        PUT FILE(SYSPRINT) EDIT('HOME PHONE',PERM_PHONE)(SKIP(2)
          ,COL(32),A,COL(46),A);
        /*

        PRINT THE STUDENT'S LOCAL ADDRESS.
                                                              */
        PUT FILE(SYSPRINT) EDIT('LOCAL ADDR.')(COL(32),A);
        LOCAL_LENGTH = INDEX(LOCAL_ADDR.CITY,'  ');
        IF LOCAL_LENGTH = 0 THEN LOCAL_LENGTH = 20;
        PUT FILE(SYSPRINT) EDIT(LOCAL_ADDR.STREET,
          SUBSTR(LOCAL_ADDR.CITY,1,LOCAL_LENGTH
          -1),', ',LOCAL_ADDR.STATE,LOCAL_ADDR.ZIP)(COL(46),A,
          COL(46),3 A,X(1),A);
        /*

        PRINT THE STUDENT'S LOCAL PHONE NUMBER.
                                                              */
```

```
PUT FILE(SYSPRINT) EDIT('LOCAL PHONE',LOCAL_PHONE)
  (SKIP(2),COL(32),A,COL(46),A);
/*

PRINT THE STUDENT'S SEX.
                                                          */
IF SEX = 'M' THEN SEX_NAME = 'MALE';
  ELSE SEX_NAME = 'FEMALE';
PUT FILE(SYSPRINT) EDIT('SEX',SEX_NAME)(SKIP(2),COL(32),
  A,COL(46),A);
/*

PRINT THE STUDENT'S BIRTH DATE.
                                                          */
PUT FILE(SYSPRINT) EDIT('BIRTHDAY',BIRTH_DATE.MONTH,'/',
  BIRTH_DATE.DAY,'/',BIRTH_DATE.YR)(SKIP(2),COL(32),A,
  COL(46),3 (F(2),A));
/*

PRINT THE STUDENT'S MARITAL STATUS.
                                                          */
PUT FILE(SYSPRINT) EDIT('MARITAL STATE',
  MARITAL_STAT_DICT(MARITAL_STAT))(SKIP(2),COL(32),A,
  COL(46),A);
/*

PRINT HOW MANY DEPENDENT CHILDREN THIS STUDENT HAS.
                                                          */
PUT FILE(SYSPRINT) EDIT('DEP. CHILDREN',DEP_CHILDREN)
  (SKIP(2),COL(32),A,COL(46),F(2));
/*

PRINT THE AGE RANGE OF THE STUDENT'S DEPENDENT CHILDREN.
                                                          */
PUT FILE(SYSPRINT) EDIT('AGE RANGE','YOUNGEST = ',
  AGE_RANGE(1),'OLDEST = ',AGE_RANGE(2))(SKIP(2),COL(32)
  ,A,COL(46),A,F(2),X(3),A,F(2));
/*

PRINT THE STUDENT'S WORK EXPERIENCES.
                                                          */
PUT FILE(SYSPRINT) EDIT('WORK EXP.',(WORK_EXP_DICT
  (WORK_EXP(NUM)) DO NUM = 1 TO 6))(SKIP(2),COL(32),A,
  COL(46),2 (A,X(3)),A,COL(51),3 (A,X(3)));
/*

PRINT THE STUDENT'S TRAVEL EXPERIENCES.
                                                          */
PUT FILE(SYSPRINT) EDIT('TRAVEL EXP.',(TRAVEL_EXP_DICT
  (TRAVEL_EXP(NUM)) DO NUM = 1 TO 6))(SKIP(2),COL(32),A,
  COL(46),2 (A,X(3)),A,COL(51),3 (A,X(3)));
/*

PRINT INFORMATION ABOUT THE STUDENT'S CURRENT EMPLOYMENT
.
                                                          */
PUT FILE(SYSPRINT) EDIT('EMPLOYMENT','EMPLOYER = ',
  EMPLOYER,'HOURS WORKED PER WEEK = ',HRS_PER_WK)
  (SKIP(2),COL(32),A,COL(46),2 A,X(3),A,F(2));
/*

PRINT INFORMATION ABOUT THE COLLEGES THE STUDENT
ATTENDED.
                                                          */
PUT FILE(SYSPRINT) EDIT('COLLEGES')(SKIP(2),COL(32),A);
DO NUM = 1 TO 4 WHILE(INSTITUTION(NUM) ¬= ' ');
  PUT FILE(SYSPRINT) EDIT('INSTITUTION = ',INSTITUTION
    (NUM),'FROM = ',BEGIN(NUM),'TO = ',END(NUM))(SKIP(2)
    ,COL(46),2 A,X(3),2 A,X(3),2 A);
  END;
/*

PRINT INFORMATION ABOUT THE STUDENT'S GPA.
                                                          */
PUT FILE(SYSPRINT) EDIT('GPA',GPA_DICT(GPA))(SKIP(2),
  COL(32),A,COL(46),A);
/*

PRINT THE STUDENT'S MYERS BRIGGS TYPE.
                                                          */
PUT FILE(SYSPRINT) EDIT('MYERS BRIGGS',MYERS_BRIGGS_DICT
  (MYERS_BRIGGS))(SKIP(2),COL(32),A,COL(46),A);
/*
```

```
PRINT THE STUDENT'S CEP ENTRANCE DATE.
                                                              */
PUT FILE(SYSPRINT) EDIT('CEP_ENTRANCE','YEAR = ',
  CEP_ENTRANCE.YR,'QUARTER = ',QUARTER_DICT
  (CEP_ENTRANCE.QTR),'HOURS COMPLETED = ',HRS_COM)
  (SKIP(2),COL(32),A,COL(46),2 A,X(3),2 A,X(3),A,F(5,1))
  ;
/*

PRINT THE STUDENT'S AREA OF SPECIALIZATION.
                                                              */
PUT FILE(SYSPRINT) EDIT('CONCENTRATION',
  SPECIALIZATION_DICT(SPECIALIZATION))(SKIP(2),COL(32),A
  ,COL(46),A);
/*

PRINT THE STUDENT'S TEAM.
                                                              */
PUT FILE(SYSPRINT) EDIT('TEAM',TEAM)(SKIP(2),COL(32),A,
  COL(46),F(1));
/*

PRINT THE NAME OF THE STUDENT'S SEMINAR LEADER.
                                                              */
PUT FILE(SYSPRINT) EDIT('SEMINAR',FACULTY(FACULTY_NO
  (TEAM,SEMINAR)))(SKIP(2),COL(32),A,COL(46),A);
/*

PRINT THE STUDENT'S EXPECTED DATE OF GRADUATION.
                                                              */
PUT FILE(SYSPRINT) EDIT('GRAD. DATE','YEAR = ',
  EXP_GRAD_DATE.YR,'QUARTER = ',QUARTER_DICT
  (EXP_GRAD_DATE.QTR))(SKIP(2),COL(32),A,COL(46),A,F(2),
  X(3),2 A);
/*

PRINT INFORMATION ABOUT THE STUDENT'S REQUIRED COURSES.
                                                              */
PUT FILE(SYSPRINT) EDIT('REQ. COURSES')(SKIP(2),COL(32),A);
/*

PRINT EXPLANATORY INFORMATION ABOUT THE GRADE.
                                                              */
PUT FILE(SYSPRINT) EDIT('GRADE = * INDICATES COURSE',
  ' REQUIREMENT HAS BEEN MET')(COL(46),2 A);
DO NUM = 1 TO 9;
  PUT FILE(SYSPRINT) EDIT(REQ_COURSE_DICT(NUM),
    'GRADE = ',REQ_COURSE_GRADE(NUM))(COL(46),A,X(4),
    2 A);
  END;
/*

PRINT INFORMATION ABOUT THE STUDENT'S REGISTRATIONS.
                                                              */
PUT FILE(SYSPRINT) EDIT('REGISTRATION')(SKIP(2),COL(32),A);
DO NUM = 1 TO 8 WHILE(REGISTRATION(NUM).YR ¬= 0);
  PUT FILE(SYSPRINT) EDIT('YR = ',REGISTRATION(NUM).YR,
    'QTR = ',REGISTRATION(NUM).QTR)(COL(46),A,F(2),X(3),
    A,F(1));
  DO KOUNT = 1 TO 5 WHILE(COURSE_NO(NUM,KOUNT) ¬= ' ');
    PUT FILE(SYSPRINT) EDIT(COURSE_NO(NUM,KOUNT),
      'HRS = ',HRS(NUM,KOUNT),'GRADE = ',REGISTRATION
      (NUM).GRADE(KOUNT))(COL(66),A,X(4),A,F(2),X(2),2 A
      );
    END;
  END;
/*

PRINT ALL INFORMATION CONTAINED IN THE STRUCTURE
PERSONAL ABOUT LEARNING ACTIVITIES THE STUDENT HAS
COMPLETED.
                                                              */
PUT FILE(SYSPRINT) EDIT('L.A.')(SKIP(2),COL(32),A);
/*

INITIALIZE ACTIVITY# AND AREA#.
                                                              */
ACTIVITY#,AREA# = 1;
DO NUM = 1 TO 98;
  /*

  IF FACULTY IS ZERO, DO NOT PRINT THIS ACTIVITY.
                                                              */
  IF LEARNING_A(NUM).FACULTY = 0 THEN GO TO CHECK;
```

```
         /*

         SET THE CORRECT GROUP AND SUBSTITUTE VALUES.
                                                              */
         IF LEARNING_A(NUM).GROUP = '1'B THEN G_STATUS = 1;
           ELSE G_STATUS = 2;
         IF LEARNING_A(NUM).SUBSTITUTE = 'I'B THEN S_STATUS = 1
           ;
           ELSE S_STATUS = 2;
         PUT FILE(SYSPRINT) EDIT('AREA = ',AREA_DICT(AREA#),
           'NO = ',ACTIVITY#,SUBSTR(LEARNING_A(NUM).DATE,1,2)
           || '/' || SUBSTR(LEARNING_A(NUM).DATE,3,2) || '/' ||
           SUBSTR(LEARNING_A(NUM).DATE,5,2),FACULTY(LEARNING_A
           (NUM).FACULTY),ACTIVITY_TITLE_DICT(ACTIVITY_TITLE
           (NUM)),GP(G_STATUS),SUB(S_STATUS))(COL(46),
           2 A,X(3),A,F(2),X(3),A,X(2),A,COL(51),2 (A,X(5)),
           COL(SI),A);
         PUT FILE(SYSPRINT) EDIT('COMMENTS =')(COL(51),A);
         DO N = 1 TO 6 WHILE(COMMENT(NUM,N) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(COMMENT_DICT(COMMENT(NUM,N))
             )(COL(56),A);
           END;
         /*

         INCREMENT ACTIVITY# BY ONE.
                                                              */
CHECK:   ACTIVITY# = ACTIVITY# + 1;
         /*

         IF THE VALUE OF THE NEW INDEX INTO LEARNING_A IS EQUAL
         TO THE VALUE OF THE NEXT ELEMENT IN ENTRY, INCREMENT
         THE AREA# BY ONE AND RESET ACTIVITY# TO ONE.
                                                              */
         IF (ACTIVITY# + ENTRY(AREA#)) = ENTRY(AREA# + 1) + 1
           THEN DO;
           AREA# = AREA# + 1;
           ACTIVITY# = 1;
           END;
        END;
       /*

       PRINT ALL INFORMATION CONTAINED IN THE STRUCTURE
       PERSONAL ABOUT FIELD EXPERIENCES THE STUDENT HAS
       COMPLETED.
                                                              */
       PUT FILE(SYSPRINT) EDIT('FIELD')(SKIP(2),COL(32),A);
       DO NUM = I TO 7 WHILE(FIELD.SCHOOL(NUM) ¬= 0);
         PUT FILE(SYSPRINT) EDIT('YR = ',FIELD(NUM).YR,'QTR = '
           ,FIELD(NUM).QTR,SCHOOL_DICT(FIELD(NUM).SCHOOL),
           FIELD(NUM).TEACHER,'GRADE = ',GRADE_DICT(FIELD(NUM).
           GRADE),CLASSROOM_DICT(FIELD(NUM).CLASSROOM),
           LEVEL_DICT(FIELD(NUM).LEVEL))(COL(46),A,F(2),X(3),
           A,F(1),X(3),A,X(1),A,COL(51),A,A,X(5),A,X(3),A);
         PUT FILE(SYSPRINT) EDIT('ROLE =')(COL(51),A);
         DO N = 1 TO 6 WHILE(F_ROLE(NUM,N) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(F_ROLE_DICT(F_ROLE(NUM,N)))
             (COL(66),A);
           END;
         PUT FILE(SYSPRINT) EDIT('STRENGTHS = ')(COL(51),A);
         DO N = 1 TO 6 WHILE(STRENGTHS (NUM,N) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(STRENGTHS_DICT(STRENGTHS
             (NUM,N)))(COL(66),A);
           END;
         PUT FILE(SYSPRINT) EDIT('NEEDS =')(COL(SI),A);
         DO N = 1 TO 6 WHILE(WEAKNESSES(NUM,N) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(WEAKNESSES_DICT(WEAKNESSES
             (NUM,N)))(COL(66),A);
           END;
         END;
       /*

       PRINT ALL INFORMATION CONTAINED IN THE STRUCTURE
       PERSONAL ABOUT THE STUDENT'S COMMUNITY EXPERIENCE.
                                                              */
       PUT FILE(SYSPRINT) EDIT('COMMUNITY')(SKIP(2),COL(32),A);
       IF COMMUNITY.AGENCY = ' ' THEN RETURN;
       PUT FILE(SYSPRINT) EDIT('YR = ',COMMUNITY.YR,'QTR = ',
         COMMUNITY.QTR,COMMUNITY.AGENCY,'AGENCY REP = ',
         AGENCY_REP,LEVEL_DICT(COMMUNITY.LEVEL))(COL(46),
         A,F(2),X(3),A,F(1),X(3),A,X(5),A,X(3),A,COL(46),A);
       PUT FILE(SYSPRINT) EDIT('ROLE =')(COL(51),A);
       DO N = I TO 4 WHILE(C_ROLE(N) ¬= 0);
         PUT FILE(SYSPRINT) EDIT(C_ROLE_DICT(C_ROLE(N)))
           (COL(66),A);
```

```
          END;
        PUT FILE(SYSPRINT) EDIT('COMMENTS =')(COL(51),A);
        DO N = 1 TO 4 WHILE(C_COMMENT(N) ¬= 0);
          PUT FILE(SYSPRINT) EDIT(C_COMMENT_DICT(C_COMMENT(N)))
          (COL(66),A);
          END;
        END;
        END;
FINISH: END;
```

```
BASIC:  PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:       BASIC

        OUTPUT
        DESCRIPTION:   LISTING OF PERSONAL INFORMATION ON
                       STUDENTS WHICH IS TO BE CHECKED AND
                       CORRECTED BY THE STUDENTS AT THE
                       BEGINNING OF EACH QUARTER.

        ALGORITHM: '   READ THE INITIAL DATA.  PROCESS THE MAIN
                       FILE BY PRINTING PERSONAL INFORMATION
                       ON STUDENTS IN SEMINARS REQUESTING
                       INFORMATION ONE SEMINAR AT A TIME.
                       REWIND THE TAPE AND CONTINUE PROCESSING
                       THE REMAINING REQUESTS BY TEAM IN A
                       SIMILAR MANNER.

        ******************************************************

        NOTE           THE FOLLOWING DECLARE STATEMENT(S)
                       INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                       ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                            1.  THE STATEMENT LABELED ONE WHICH
                                CONTAINS THE STRUCTURE PERSONAL.
                            2.  THE STATEMENT LABELED TWO WHICH
                                CONTAINS THE DICTIONARIES.

        ******************************************************

                                                              */
        /*

        READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
        USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
        FACULTY_NO.
                                                              */
        GET FILE(SYSIN) EDIT(MIN,MAX)(COL(1),2 (F(1),X(4)));
        BEGIN;
        DECLARE
          /*
          BIT WHEN SET TO A '1'B INDICATES THAT PRINTING
          FOR THIS SEMINAR HAS BEEN COMPLETED.                */
          BIT BIT(1) INITIAL('0'B),
          /*
          DICTIONARY OF FACULTY MEMBER'S NAMES ASSOCIATED WITH
          THE CODES RECORDED IN PERSONAL.                     */
          FACULTY(25) CHARACTER(14) INITIAL('BERNARD','CURRAN',
             'DETURE','FEY','GOLDMAN','HILLIARD','JESTER',
             'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
             'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
             'TISON','WILMOT','KRALL','CREEDON','LEVINE',
             'HEDGES','GRIMM'),
          /*
          FACULTY NUMBERS OF THE SEMINAR LEADERS BY TEAM.     */
          FACULTY_NO(MIN:MAX,4) FIXED BINARY,
          CUR_DATE CHARACTER(8),
          MARK(2) CHARACTER(2),
          OLD FILE RECORD SEQUENTIAL,
          RETURN_DATE CHARACTER(15),
          SEMINAR# FIXED BINARY,
          TEAM# FIXED BINARY;
        /*

        INITIALIZE FACULTY_NO.
                                                              */
        FACULTY_NO = 0;
        /*

        ON ENCOUNTERING AN END OF FILE ON OLD, REWIND THE TAPE
        AND SET BIT TO A '1'B.
                                                              */
        ON ENDFILE(OLD) BEGIN;
          CLOSE FILE(OLD);
          OPEN FILE(OLD);
          BIT = '1'B;
          END;
        OPEN FILE(SYSPRINT) LINESIZE(130);
        /*

        READ THE CURRENT DATE.  THEN READ THE DATE THE CORRECTED
        FORM SHOULD BE RETURNED.
                                                              */
          GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
          GET FILE(SYSIN) EDIT(RETURN_DATE)(COL(1),A(15));
          /*

          READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
          TEAM READ THE TEAM NUMBER, THE NUMBER OF SEMINARS TO BE
          PROCESSED, AND THE SEMINAR NUMBER AND FACULTY NUMBER
          OF EACH SEMINAR TO BE PROCESSED.
                                                                     */
          GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
          DO I = 1 TO NO_CARDS;
            GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
              TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
              X(4)),4 (F(1),X(4),F(2),X(3)));
            END;
          /*

          PRINT PERSONAL INFORMATION ABOUT STUDENTS IN EACH
          SEMINAR BY TEAM.  PRINT ONE PER PAGE.
                                                                     */
          DO TEAM# = MIN TO MAX;
            IF (FACULTY_NO(TEAM#,1) = 0 & FACULTY_NO(TEAM#,2) = 0
              & FACULTY_NO(TEAM#,3) = 0 & FACULTY_NO(TEAM#,4) = 0)
              THEN GO TO T_END;
            /*

            PROCESS ONE SEMINAR AT A TIME.
                                                                     */
            DO SEMINAR# = 1 TO 4;
              IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO S_END;
              /*

              PRINT THE HEADER INFORMATION FOR EACH SEMINAR.
                                                                     */
              PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
              PUT FILE(SYSPRINT) EDIT('BASIC','OUTPUT DESCRIPTION'
                ,': LISTING OF PERSONAL INFORMATION ON STUDENTS '
                ,'WHICH IS TO CHECKED AND CORRECTED BY THE ',
                'STUDENTS AT THE BEGINNING OF EACH QUARTER.')(SKIP
                ,A,SKIP,4 A);
              /*

              PRINT THE RETURN DATE.
                                                                     */
              PUT FILE(SYSPRINT) EDIT('RETURN DATE: ',RETURN_DATE)
                (SKIP,2 A);
              /*

           PRINT THE SEMINAR LEADER'S NAME.
                                                                     */
              PUT FILE(SYSPRINT) EDIT('SEMINAR LEADER: ',FACULTY
                (FACULTY_NO(TEAM#,SEMINAR#)))(SKIP,2 A);
              /*

              READ A RECORD FROM THE MAIN FILE.
                                                                     */
LOOP:         READ FILE(CLD) INTO(PERSONAL);
              /*

              IF THE END OF THE MAIN FILE HAS BEEN ENCOUNTERED,
              INDICATED BY BIT SET TO A '1'B, RESET BIT TO A '0'B
              AND TRANSFER TO S_END TO BEGIN PROCESSING A NEW
              SEMINAR.
                                                                     */
              IF BIT = '1'B THEN DO;
                BIT = '0'B;
                GO TO S_END;
                END;
              /*

              CHECK IF THIS STUDENT IS IN THE TEAM AND SEMINAR
              CURRENTLY BEING PRINTED.
                                                                     */
              IF TEAM = TEAM# & SEMINAR = SEMINAR# THEN DO;
                /*

                PRINT THE DIRECTIONS TO THE STUDENT.
                                                                     */
                PUT FILE(SYSPRINT) EDIT('DIRECTIONS: PLEASE ',
                  'CORRECT ANY INFORMATION BELOW WHICH IS ',
                  'INCORRECT.','RETURN THIS FORM TO YOUR SEMINAR',
                  ' LEADER NO LATER THEN ',RETURN_DATE,'.')(PAGE,
                  3 A,COL(5),4 A);
                /*
```

```
                    PRINT THE STUDENT'S SOCIAL SECURITY NUMBER.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('SOCIAL SECURITY NUMBER',
                      '(1) : ',SOC_SEC_NO)(SKIP(4),2 A,COL(40),A);
                    /*
                    PRINT THE STUDENT'S NAME.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('NAME(2) : ','LAST(3) : ',
                      LAST,'FIRST(4) : ',FIRST,'MIDDLE(5) : ',MIDDLE)
                      (SKIP(2),A,3 (COL(10),A,COL(40),A));
                    /*
                    PRINT THE STUDENT'S PERMANENT ADDRESS.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('PERMANENT ADDRESS(6) : ',
                      'STREET(7) : ',PERM_ADDR.STREET,'CITY(8) : ',
                      PERM_ADDR.CITY,'STATE(9) : ',PERM_ADDR.STATE,
                      'ZIP CODE(10) : ',PERM_ADDR.ZIP)(SKIP(2),A,
                      4 (COL(10),A,COL(40),A));
                    /*
                    PRINT THE STUDENT'S PERMANENT PHONE NUMBER.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('PERMANENT PHONE NUMBER',
                      '(11) : ',PERM_PHONE)(SKIP(2),2 A,COL(40),A);
                    /*
                    PRINT THE STUDENT'S LOCAL ADDRESS.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('LOCAL ADDRESS(12) : ',
                      'STREET(13) : ',LOCAL_ADDR.STREET,'CITY(14) : ',
                      LOCAL_ADDR.CITY,'STATE(15) : ',LOCAL_ADDR.STATE,
                      'ZIP CODE(16) : ',LOCAL_ADDR.ZIP)(SKIP(2),A,
                      4 (COL(10),A,COL(40),A));
                    /*
                    PRINT THE STUDENT'S LOCAL PHONE NUMBER.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('LOCAL PHONE NUMBER(17)',
                      ' : ',LOCAL_PHONE)(SKIP(2),2 A,COL(40),A);
                    /*
                    PRINT THE DIFFERENT MARITAL STATUS TYPES.  PLACE
                    AN 'X' NEXT TO THE STUDENT'S CURRENT MARITAL
                    STATUS.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('MARITAL STATUS(23) : ',
                      'PLEASE CHECK YOUR CURRENT MARITAL STATUS.')
                      (SKIP(2),A,COL(40),A);
                    DO I = 1 TO 4;
                      PUT FILE(SYSPRINT) EDIT(MARITAL_STAT_DICT(I))
                        (SKIP,COL(45),A);
                      IF MARITAL_STAT = I THEN PUT FILE(SYSPRINT) EDIT
                        ('X')(COL(75),A);
                      PUT FILE(SYSPRINT) EDIT('(',I,')')(COL(78),A,
                        F(2),A);
                      END;
                    /*
                    PRINT HOW MANY DEPENDENT CHILDREN THIS STUDENT HAS
                    .
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('NUMBER OF DEPENDENT ',
                      'CHILDREN(24) : ',DEP_CHILDREN)(SKIP(2),2 A,
                      COL(40),F(2));
                    IF DEP_CHILDREN = 0 THEN GO TO NEXT;
                    /*
                    PRINT THE AGE RANGE OF THE STUDENT'S DEPENDENT
                    CHILDREN.
                                                                 */
                    PUT FILE(SYSPRINT) EDIT('AGE RANGE OF DEPENDENT ',
                      'CHILDREN(26) : ','YOUNGEST CHILD : ',
                      AGE_RANGE(1),'OLDEST CHILD : ',AGE_RANGE(2))
                      (SKIP(2),2 A,2 (COL(10),A,COL(40),A));
                    /*
                    PRINT THE DIFFERENT TYPES OF WORK EXPERIENCES.
                    PLACE AN 'X' NEXT TO EACH OF THE STUDENT'S
                    PREVIOUS WORK EXPERIENCES.
                                                                 */
NEXT:               PUT FILE(SYSPRINT) EDIT('WORK EXPERIENCE(26) : ',
                'PLEASE CHECK THE TYPES OF WORK EXPERIENCES YOU'
                ,' HAVE HAD.')(SKIP(2),A,COL(40),2 A);
              DO I = I TO 9;
                MARK = ' ';
                DO INDEX = I TO 6;
                  IF WORK_EXP(INDEX) = I THEN
                    DO;
                      MARK(I) = 'X';
                      GO TO CHECK_1;
                      END;
                  END;
CHECK_1:        DO INDEX = I TO 6;
                  IF WORK_EXP(INDEX) = I + 9 THEN
                    DO;
                      MARK(2) = 'X';
                      GO TO PRINT_I;
                      END;
                  END;
PRINT_I:        PUT FILE(SYSPRINT) EDIT(WORK_EXP_DICT(I),MARK(1)
                  ,'(',I,')',WORK_EXP_DICT(I + 9),MARK(2),'(',
                  I + 9,')')(SKIP,COL(45),A,COL(75),A,COL(78),A,
                  F(2),A,COL(90),A,COL(120),A,COL(123),A,F(2),A)
                  ;
                END;
              /*

              PRINT THE DIFFERENT TYPES OF TRAVEL EXPERIENCES.
              PLACE AN 'X' NEXT TO EACH OF THE STUDENT'S
              PREVIOUS TRAVEL EXPERIENCES.
                                                                 */
              PUT FILE(SYSPRINT) EDIT('TRAVEL EXPERIENCE(28) : '
                ,'PLEASE CHECK THE TYPES OF TRAVEL EXPERIENCES '
                ,'YOU HAVE HAD.')(SKIP(2),A,COL(40),2 A);
              DO I = 1 TO 10;
                MARK = ' ';
                DO INDEX = I TO 6;
                  IF TRAVEL_EXP(INDEX) = I THEN
                    DO;
                      MARK(1) = 'X';
                      GO TO CHECK_2;
                      END;
                  END;
CHECK_2:        DO INDEX = 1 TO 6;
                  IF TRAVEL_EXP(INDEX) = I + 10  THEN
                    DO;
                      MARK(2) = 'X';
                      GO TO PRINT_2;
                      END;
                  END;
PRINT_2:        PUT FILE(SYSPRINT) EDIT(TRAVEL_EXP_DICT(I),
                  MARK(1),'(',I,')',TRAVEL_EXP_DICT(I + 10),
                  MARK(2),'(',I + 10,')')(SKIP,COL(45),A,COL(75)
                  ,A,COL(78),A,F(2),A,COL(90),A,COL(120),A,
                  COL(123),A,F(2),A);
                END;
              /*

              PRINT INFORMATION ABOUT THE STUDENT'S CURRENT
              EMPLOYMENT.
                                                                 */
              PUT FILE(SYSPRINT) EDIT('CURRENT EMPLOYMENT(30) :'
                ,' ','EMPLOYER(31) : ',EMPLOYER,
                'HOURS WORKED PER WEEK(32) : ',HRS_PER_WK)
                (SKIP(2),2 A,COL(10),A,COL(40),A,COL(10),A,
                COL(40),F(2));
              /*

              PRINT THE STUDENT'S MYERS BRIGGS TYPE.
                                                                 */
              PUT FILE(SYSPRINT) EDIT('MYERS BRIGGS TYPE(39) : '
                ,MYERS_BRIGGS_DICT(MYERS_BRIGGS))(SKIP(2),A,
                COL(40),A);
              /*
              PRINT THE STUDENT'S AREA OF SPECIALIZATION.
                                                                 */
              PUT FILE(SYSPRINT) EDIT('SPECIALIZATION(44) : ',
                SPECIALIZATION_DICT(SPECIALIZATION))(SKIP(2),A,
                COL(40),A);
              /*

              PRINT THE STUDENT'S TEAM NUMBER.
                                                                 */
              PUT FILE(SYSPRINT) EDIT('TEAM(45) : ',TEAM)
                (SKIP(2),A,COL(40),F(2));
```

```
                 /*
                 PRINT THE NAME OF THE STUDENT'S SEMINAR LEADER.
                                                                 */
                 PUT FILE(SYSPRINT) EDIT('SEMINAR(46) : ',FACULTY(
                   FACULTY_NO(TEAM#,SEMINAR#)))(SKIP(2),A,COL(40),
                   A);
                 /*
                 PRINT THE STUDENT'S EXPECTED DATE OF GRADUATION.
                                                                 */
                 PUT FILE(SYSPRINT) EDIT('EXPECTED DATE OF ',
                   'GRADUATION(47) : ','YEAR(48) : ',
                   EXP_GRAD_DATE.YR,'QUARTER(49) : ',QUARTER_DICT
                   (EXP_GRAD_DATE.QTR))(SKIP(2),2 A,COL(10),A,
                   COL(40),F(2),COL(10),A,COL(40),A);
                 END;
                 /*
                 TRANSFER TO LOOP TO PROCESS A NEW RECORD.
                                                                 */
                 GO TO LOOP;
S_END:         END;
T_END:       END;
           END;
FINISH:    END;
```

```
BUILD:   PROCEDURE OPTIONS(MAIN);
         /*

         PROGRAM:          BUILD

         OUTPUT
         DESCRIPTION:      A LIST BY SOCIAL SECURITY NUMBER OF ALL
                           DATA ENTERED DURING THE ADDITION OF NEW
                           RECORDS.

         ALGORITHM:        READ THE INITIAL DATA.  FOR EACH NEW
                           STUDENT RECORD, INITIALIZE CERTAIN ITEMS
                           IN THE STRUCTURE PERSONAL.  READ OTHER
                           INITIAL VALUES, PRINT THESE VALUES AND
                           STORE THE RECORD IN THE NEW MAIN FILE.
                           WHEN THERE ARE NO MORE NEW RECORDS CHECK
                           THE VALUE OF TYPE.  IF TYPE IS SET TO 1,
                           COPY THE OLD MAIN FILE ONTO THE NEW MAIN
                           FILE.

         ************************************************************

         NOTE              THE FOLLOWING DECLARE STATEMENT(S)
                           INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                           ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                                1.   THE STATEMENT LABELED ONE WHICH
                                     CONTAINS THE STRUCTURE PERSONAL.

         ************************************************************

                                                                   */
         /*

         ON ENCOUNTERING AN END OF FILE ON SYSIN, TRANSFER TO
         THE LABEL CHECK TO DETERMINE IF RECORDS FROM THE OLD
         MAIN FILE SHOULD BE COPIED ONTO THE NEW MAIN FILE.
                                                                   */
         ON ENDFILE(SYSIN) GO TO CHECK;
         DECLARE
           /*
           FLAG WHEN SET TO A '*' INDICATES THE END OF A
           STUDENT RECORD.                                         */
           FLAG CHARACTER(1),
           CUR_DATE CHARACTER(8),
           NEW FILE RECORD SEQUENTIAL,
           OLD FILE RECORD SEQUENTIAL,
           TYPE FIXED BINARY;
         SUB_AREA = 0;
         CTR_QUESTIONNAIRE = 0;
         OPEN FILE(SYSPRINT) LINESIZE(130);
         /*

         ON ENCOUNTERING THE END OF A PAGE OF PRINT, PRINT THE
         HEADER INFORMATION.
                                                                   */
         ON ENDPAGE(SYSPRINT) BEGIN;
           PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
           PUT FILE(SYSPRINT) EDIT('BUILD',
             'OUTPUT DESCRIPTION: A LISTING BY SOCIAL SECURITY',
             ' NUMBER OF ALL DATA ENTERED DURING THE ADDITION OF'
             ,' NEW RECORDS')(SKIP,A,SKIP,3 A);
           PUT FILE(SYSPRINT) EDIT('SOCIAL SECURITY NO','ITEM',
             'PRESENT CONTENTS')(SKIP(6),A,COL(25),A,COL(35),A);
           PUT FILE(SYSPRINT) EDIT('__________________','____',
             '________________')(SKIP(0),A,COL(25),A,COL(35),A);
           END;
         /*

         ON ENCOUNTERING A CONVERSION ERROR PRINT AN ERROR
         MESSAGE.  CONTINUE TO READ THE REMAINING CARDS IN THIS
         RECORD.  DO NOT INCLUDE THIS RECORD IN THE MAIN FILE.
         ON ENCOUNTERING A '*' TRANSFER TO THE LABEL LOOP TO
         BEGIN PROCESSING A NEW RECORD.
                                                                   */
         ON CONVERSION BEGIN;
           PUT FILE(SYSPRINT) EDIT('** CONVERSION **')(SKIP,COL
             (26),A);
NEXT:      GET FILE(SYSIN) EDIT(FLAG)(SKIP,A(1));
           IF FLAG = '*' THEN GO TO LOOP;
           GO TO NEXT;
           END;
         /*

         ON ENCOUNTERING THE END OF THE OLD MAIN FILE TRANSFER TO
```

```
          FINISH.
                                                                    */
          ON ENDFILE(OLD) GO TO FINISH;
          /*
          FIRST READ THE VALUE OF TYPE AND THEN READ THE CURRENT
          DATE.
                                                                    */
          GET FILE(SYSIN) EDIT(TYPE)(COL(1),F(1));
          GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
          /*

          PRINT THE HEADER INFORMATION FOR THE FIRST PAGE OF
          PRINT.
                                                                    */
          PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
          PUT FILE(SYSPRINT) EDIT('BUILD',
            'CUTPUT DESCRIPTICN: A LISTING BY SOCIAL SECURITY',
            ' NUMBER OF ALL DATA ENTERED DURING THE ADDITION OF ',
            'NEW RECORDS')(SKIP,A,SKIP,3 A);
          PUT FILE(SYSPRINT) EDIT('SOCIAL SECURITY NO','ITEM',
            'PFESENT CONTENTS')(SKIP(6),A,CCL(25),A,COL(35),A);
          PUT FILE(SYSPRINT) EDIT('_________________','____',
            '________________')(SKIP(0),A,COL(25),A,COL(35),A);
LOOP:
          /*

          INITIALIZE CERTAIN CATA ITEMS IN THE STRUCTURE PERSONAL.
                                                                    */
          REQ_COURSE_GRADE = ' ';
          CCL_ATT = ' ';
          CONFLETION_CATE = 0;
          /*

          READ OTHER INITIAL VALUES FOR DATA ITEMS IN THIS
          STUDENT'S RECORD.  AFTER READING EACH INITIAL VALUE,
          PRINT THE VALUE WHICH IS STORED IN PERSONAL.
                                                                    */
          GET FILE(SYSIN) EDIT(SOC_SEC_NC)(COL(1),A(9));
          PUT FILE(SYSPRINT) EDIT(SOC_SEC_NO)(SKIP(2),X(4),A);
          GET FILE(SYSIN) EDIT(ITEM,NAME)(SKIP,F(2),X(3),A(14),
            A(1),A(10),X(5),A(14));
          PUT FILE(SYSPRINT) EDIT(ITEM.NAME)(COL(26),F(2),X(7),
            3 (A,X(2)));
          GET FILE(SYSIN) EDIT(ITEM,PERM_ADDR)(SKIP,F(2),X(3),
            A(20),X(5),A(20),X(5),A(2),X(3),A(5));
          PUT FILE(SYSPRINT) EDIT(ITEM,PERM_ADDR)(COL(26),F(2),
            X(7),4 (A,X(2)));
          GET FILE(SYSIN) EDIT(ITEM,PERM_PHONE)(SKIP,F(2),X(3),
            A(10));
          PUT FILE(SYSPRINT) EDIT(ITEM,PERM_PHONE)(COL(26),F(2),
            X(7),A);
          GET FILE(SYSIN) EDIT(ITEM,LOCAL_ADDR)(SKIP,F(2),X(3),
            A(20),X(5),A(20),X(5),A(2),X(3),A(5));
          PUT FILE(SYSPRINT) EDIT(ITEM,LOCAL_ADDR)(COL(26),F(2),
            X(7),4 (A,X(2)));
          GET FILE(SYSIN) EDIT(ITEM,LOCAL_PHONE)(SKIP,F(2),X(3),
            A(7));
          PUT FILE(SYSPRINT) EDIT(ITEM.LOCAL_PHONE)(COL(26),F(2),
            X(7),A);
          GET FILE(SYSIN) EDIT(ITEM,SEX)(SKIP,F(2),X(3),A(1));
          PUT FILE(SYSPRINT) ECIT(ITEM,SEX)(COL(26),F(2),X(7),A);
          GET FILE(SYSIN) EDIT(ITEM,BIRTH_DATE)(SKIP,F(2),X(3),3
            (F(2),X(3)));
          PUT FILE(SYSPRINT) EDIT(ITEM,BIRTH_DATE)(COL(26),F(2),
            X(7),3 (F(2),X(2)));
          GET FILE(SYSIN) EDIT(ITEM,MARITAL_STAT)(SKIP,F(2),X(3),
            F(1));
          PUT FILE(SYSPRINT) ECIT(ITEM,MARITAL_STAT)(CCL(26),F(2),
            X(7),F(1));
          GET FILE(SYSIN) EDIT(ITEM,DEP_CHILDREN)(SKIP,F(2),X(3),
            F(1));
          PUT FILE(SYSPRINT) EDIT(ITEM,DEP_CHILDREN)(COL(26),F(2),
            X(7),F(1));
          GET FILE(SYSIN) EDIT(ITEM,AGE_RANGE)(SKIP,F(2),X(3),2 (
            F(2),X(3)));
          PUT FILE(SYSPRINT) EDIT(ITEM,AGE_RANGE)(CCL(25),F(2),
            X(7),2 (F(2),X(2)));
          GET FILE(SYSIN) EDIT(ITEM,WORK_EXP)(SKIP,F(2),X(3),
            6(F(2),X(3)));
          PUT FILE(SYSPRINT) EDIT(ITEM,WORK_EXP)(COL(26),F(2),
            X(7),6(F(2),X(2)));
          GET FILE(SYSIN) EDIT(ITEM,TRAVEL_EXP)(SKIP,F(2),X(3),
            6(F(2),X(3)));
          PUT FILE(SYSPRINT) EDIT(ITEM.TRAVEL_EXP)(CCL(26),F(2),
          X(7),6(F(2),X(2)));
        GET FILE(SYSIN) EDIT(ITEM,CUR_EMPLOYMENT)(SKIP,F(2),
          X(3),A(20),X(5),F(2));
        PUT FILE(SYSPRINT) EDIT(ITEM,CUR_EMPLOYMENT)(COL(26),
          F(2),X(7),A,X(2),F(2));
        GET FILE(SYSIN) EDIT(ITEM,NO_COL_ATT,(COL_ATT(I) DO I =
          1 TO NO_COL_ATT))(SKIP,F(2),X(3),F(1),(NO_COL_ATT)
          (COL(11),A(30),X(5),2 (A(6),X(4)))));
        PUT FILE(SYSPRINT) EDIT(ITEM,(COL_ATT(I) DO I = 1 TO
          NO_COL_ATT))(COL(26),F(2),(NO_COL_ATT)(COL(35),3 (A,X
          (2)))));
        GET FILE(SYSIN) EDIT(ITEM,GPA)(SKIP,F(2),X(3),F(1));
        PUT FILE(SYSPRINT) EDIT(ITEM,GPA)(COL(26),F(2),X(7),F(1)
          );
        GET FILE(SYSIN) EDIT(ITEM,MYERS_BRIGGS)(SKIP,F(2),X(3),
          F(2));
        PUT FILE(SYSPRINT) EDIT(ITEM,MYERS_BRIGGS)(COL(26),F(2),
          X(7),F(2));
        GET FILE(SYSIN) EDIT(ITEM,CEP_ENTRANCE)(SKIP,F(2),X(3),2
          (F(2),X(3)),F(4.1));
        PUT FILE(SYSPRINT) EDIT(ITEM,CEP_ENTRANCE)(COL(26),F(2),
          X(7),2 (F(2),X(2)),F(5,1));
        GET FILE(SYSIN) EDIT(ITEM,SPECIALIZATION)(SKIP,F(2),X(3)
          ,F(1));
        PUT FILE(SYSPRINT) EDIT(ITEM,SPECIALIZATION)(COL(26),
          F(2),X(7),F(1));
        GET FILE(SYSIN) EDIT(ITEM,TEAM)(SKIP,F(2),X(3),F(1));
        PUT FILE(SYSPRINT) EDIT(ITEM,TEAM)(COL(26),F(2),X(7),
          F(1));
        GET FILE(SYSIN) EDIT(ITEM,SEMINAR)(SKIP,F(2),X(3),F(1));
        PUT FILE(SYSPRINT) EDIT(ITEM,SEMINAR)(COL(26),F(2),X(7),
          F(1));
        GET FILE(SYSIN) EDIT(ITEM,EXP_GRAD_DATE)(SKIP,F(2),X(3),
          F(2),X(3),F(1));
        PUT FILE(SYSPRINT) EDIT(ITEM,EXP_GRAD_DATE)(COL(26),F(2)
          ,X(7),2 (F(2),X(2)));
        GET FILE(SYSIN) EDIT(ITEM,REQ_COURSE_GRADE)(SKIP,F(2),
          X(3),9 (A(1),X(4)));
        PUT FILE(SYSPRINT) EDIT(ITEM,REQ_COURSE_GRADE)(COL(26),
          F(2),X(7),9(A,X(2)));
        /*

        READ THE VALUE OF FLAG.
                                                              */
        GET FILE(SYSIN) EDIT(FLAG)(SKIP,A(1));
        /*

        COPY THIS INITIALIZED RECORD ONTO THE NEW MAIN FILE.
                                                              */
        WRITE FILE(NEW) FROM(PERSONAL);
        /*

        TRANSFER TO THE LABEL LOOP TO BEGIN PROCESSING A NEW
        RECORD.
                                                              */
        GO TO LOOP;
        /*

        IF TYPE IS SET TO 1, WHICH INDICATES THAT AN OLD MAIN
        EXISTS, COPY THE RECORDS FROM THE OLD MAIN FILE ONTO
        THE NEW MAIN FILE.
                                                              */
CHECK:  IF TYPE = 1 THEN DO;
COPY:     READ FILE(OLD) INTO(PERSONAL);
          WRITE FILE(NEW) FROM(PERSONAL);
          GO TO COPY;
          END;
FINISH: END;
```

```
EXIT:   PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:          CHECK

        OUTPUT
        DESCRIPTION:      FOR EACH EXPECTED GRADUATE THERE IS A
                          LIST OF MISSING LEARNING ACTIVITIES BY
                          AREA AND THE TOTAL NUMBER OF MISSING
                          LEARNING ACTIVITIES.  IN ADDITION, THERE
                          IS AN ALPHABETICAL LIST OF EXPECTED
                          GRADUATES IN A TEAM BY SEMINAR.  THOSE
                          STUDENTS WITH MORE THAN FIVE LEARNING
                          ACTIVITIES TO COMPLETE ARE MARKED BY AN
                          '*'.

        ALGORITHM:        READ THE INITIAL DATA.  PROCESS THE MAIN
                          FILE STORING IN THE STRUCTURE GRAD
                          INFORMATION ABOUT EXPECTED GRADUATES
                          IN THE SEMINARS BEING PROCESSED.
                          ON ENCOUNTERING THE END OF THE MAIN FILE
                          SORT THE SEMINARS IN THE STRUCTURE GRAD
                          ALPHABETICALLY BY LAST NAME.  ELIMINATE
                          EXCESS BLANKS FROM THE NAMES IN THE
                          STRUCTURE GRAD.  FOR THE REQUESTS
                          MADE, PRINT THE INFORMATION ABOUT THE
                          EXPECTED GREDUATES IN TWO FORMS.

        ***************************************************

        NOTE              THE FOLLOWING DECLARE STATEMENT(S)
                          INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                          ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                               1.  THE STATEMENT LABELED ONE WHICH
                                   CONTAINS THE STRUCTURE PERSONAL.

        ***************************************************
                                                                 */
        /*

        READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
        USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
        GRAD, FACULTY_NO AND VALUE.
                                                                 */
        GET FILE(SYSIN) EDIT(MIN#,MAX#)(COL(1),2 (F(1),X(4)));
        BEGIN;
        DECLARE
          /*
          INDICATOR OF THE PRINT STATUS FOR THIS TEAM.           */
          BIT(MIN#:MAX#) BIT(1),
          /*
          AN ENTRY POINT, FOR INSTANCE ENTRY(1), INDICATES THE
          NUMBER OF THE POSITION PRECEEDING THE BEGINNING OF
          THIS SUBSTANTIVE AREA IN LEARNING_A.                   */
          ENTRY(11) FIXED BINARY INITIAL(0,8,19,31,41,50,56,66,
            82,87,95),
          /*
          DICTIONARY OF FACULTY MEMBER'S NAMES ASSOCIATED WITH
          THE CODES RECORDED IN PERSONAL.                        */
          FACULTY(20) CHARACTER(14) INITIAL('BERNARD','CURRAN',
            'DETURE','FEY','GOODMAN','HILLIARD','JESTER',
            'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
            'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
            'TISON','WILMOT'),
          /*
          FACULTY NUMBERS OF THE SEMINAR LEADERS BY TEAM.        */
          FACULTY_NO(MIN#:MAX#,4) FIXED BINARY,
          /*
          SELECTED DATA STORED ON EXPECTED GRADUATES IN EACH
          TEAM BY SEMINAR.                                       */
          1 GRAD(MIN#:MAX#,4,10),
            2 NAME CHARACTER(25),
            2 ACTIVITY(11,5) FIXED BINARY,
            2 TOTAL FIXED BINARY,
          /*
          EXPECTED GRADUATION DATE TO BE CHECKED.                */
          1 TEST_DATE,
            2 YR FIXED BINARY,
            2 QTR FIXED BINARY,
          /*
          COUNTER FOR THE NUMBER OF EXPECTED GRADUATES IN EACH
          TEAM BY SEMINAR.                                       */
          VALUE(MIN#:MAX#,4) FIXED BINARY,
           /*
           COUNTER FOR THE NUMBER OF MISSING LEARNING ACTIVITIES
           IN EACH SUBSTANTIVE AREA.                                */
           VALUE_A(11) FIXED BINARY,
           AREA(98) FIXED BINARY INITIAL((6)1,(13)2,(12)3,
              (10)4,(9)5,(6)6,(10)7,(16)8,(5)9,(8)10,(3)11),
           CUR_DATE CHARACTER(8),
           OLD FILE RECORD SEQUENTIAL,
           SEMINAR# FIXED BINARY,
           STR CHARACTER(25),
           TEAM# FIXED BINARY,
           1 TEMP,
              2 NAME CHARACTER(25),
              2 ACTIVITY(11,5) FIXED BINARY,
              2 TOTAL FIXED BINARY;
         /*

         INITIALIZE GRAD, FACULTY_NO AND VALUE.
                                                                    */
         GRAD.NAME = '0';
         GRAD.ACTIVITY = 0;
         GRAD.TOTAL = 0;
         FACULTY_NO = 0;
         VALUE = 0;
         /*

         ON ENCOUNTERING AN ILLEGAL SUBSCRIPT IN A STUDENT'S
         RECORD, PRINT THE STUDENT'S SOCIAL SECURITY NUMBER, TEAM
         NUMBER, FACULTY NUMBER, SEMINAR NUMBER AND THE TOTAL
         NUMBER OF EXPECTED GRADUATES IN THIS SEMINAR.
                                                                    */
         ON SUBSCRIPTRANGE BEGIN;
           PUT FILE(SYSPRINT) EDIT('SUBSCRIPT OUTSIDE LEGAL',
              ' BOUNDS','SOC_SEC_NO = ',SOC_SEC_NO,'TEAM = ',TEAM,
              'SEMINAR = ',SEMINAR,'VALUE = ',VALUE(TEAM,SEMINAR))
              (2 A,SKIP(3),A,A,X(5),3 (A,F(2),X(5)));
           GO TO FINISH;
           END;
         /*
         ON ENCOUNTERING THE END OF THE MAIN FILE TRANSFER TO
         SORT.                                                      */
         ON ENDFILE(OLD) GO TO SORT;
         OPEN FILE(SYSPRINT) LINESIZE(130);
         /*

         READ THE CURRENT DATE.   THEN READ THE EXPECTED
         GRADUATION DATE TO BE CHECKED.
                                                                    */
         GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
         GET FILE(SYSIN) EDIT(TEST_DATE)(COL(1),F(2),X(3),F(1));
         /*

         READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
         TEAM READ THE TEAM NUMBER, THE NUMBER OF SEMINARS TO BE
         PROCESSED, AND THE SEMINAR NUMBER AND FACULTY NUMBER
         OF EACH SEMINAR TO BE PROCESSED.
                                                                    */
         GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
         DO I = 1 TO NO_CARDS;
           GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
              TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
              X(4)),4 (F(1),X(4),F(2),X(3)));
              END;
         /*

         PROCESS THE FILE SEQUENTIALLY.  READ A STUDENT'S RECORD.
         IF THE RECORD IS NOT TO BE PRINTED, TRANSFER
         TO LOOP TO OBTAIN A NEW RECORD.   CHECK THE
         STUDENT'S EXPECTED GRADUATION DATE AGAINST THE
         TEST DATE.  IF THE DATES ARE NOT EQUAL, OBTAIN A NEW
         RECORD.  OTHERWISE, INCREMENT THE CORRECT COUNTER IN
         VALUE BY ONE.  STORE THE STUDENT'S NAME, THE NUMBERS OF
         THE MISSING LEARNING ACTIVITIES BY AREA WITH A MAXIMUM
         OF FIVE ACTIVITIES PER AREA, AND THE TOTAL NUMBER OF
         MISSING LEARNING ACTIVITIES IN THE STRUCTURE GRAD.
                                                                    */
LOOP:    READ FILE(OLD) INTO(PERSONAL);
         IF TEAM < MIN# | TEAM > MAX# THEN GO TO LOOP;
         IF FACULTY_NO(TEAM,SEMINAR) = 0 THEN GO TO LOOP;
         IF EXP_GRAD_DATE.YR ¬= TEST_DATE.YR |
             EXP_GRAD_DATE.QTR ¬= TEST_DATE.QTR THEN GO TO LOOP;
         VALUE(TEAM,SEMINAR) = VALUE(TEAM,SEMINAR) + 1;
         GRAD.NAME(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) =
           PERSONAL.LAST || ',' || PERSONAL.FIRST;
         VALUE_A = 0;
         DO I = 1 TO 98;
           IF LEARNING_A.FACULTY(I) = 0 THEN DO;
             VALUE_A(AREA(I)) = VALUE_A(AREA(I)) + 1;
             IF VALUE_A(AREA(I)) > 5 THEN GO TO NEXT;
             GRAD.ACTIVITY(TEAM,SEMINAR,VALUE(TEAM,SEMINAR),
               AREA(I),VALUE_A(AREA(I))) = I - ENTRY(AREA(I));
NEXT:        GRAD.TOTAL(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) =
               GRAD.TOTAL(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) + 1;
             END;
           END;
         GO TO LOOP;
         /*

         SET BIT TO A '1'B IF THIS TEAM IS NOT TO BE PRINTED.
         SORT THE SEMINARS IN THE STRUCTURE GRAD ALPHABETICALLY
         BY NAME.
                                                                      */
SORT:    DO TEAM# = MIN# TO MAX#;
           BIT(TEAM#) = FACULTY_NO(TEAM#,1) = 0 &
             FACULTY_NO(TEAM#,2) = 0 & FACULTY_NO(TEAM#,3) = 0 &
             FACULTY_NO(TEAM#,4) = 0;
           IF BIT(TEAM#) = '1'B THEN GO TO T_END;
           DO SEMINAR# = 1 TO 4;
             IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO S_END;
             DO I = 1 TO 9;
               DO J = I+1 TO 10;
                 IF GRAD.NAME(TEAM#,SEMINAR#,I) >
                   GRAD.NAME(TEAM#,SEMINAR#,J) THEN DO;
                   TEMP = GRAD(TEAM#,SEMINAR#,I);
                   GRAD(TEAM#,SEMINAR#,I) = GRAD(TEAM#,
                     SEMINAR#,J);
                   GRAD(TEAM#,SEMINAR#,J) = TEMP;
                   END;
               END;
             END;
S_END:     END;
T_END:   END;
         /*

         PROCESS THE NAMES IN THE STRUCTURE GRAD.  CHANGE ALL
         DUMMY NAMES, NAMES WHICH HAD BEEN INITIALIZED TO '0'
         FOR SORTING AND HAVE NOT BEEN CHANGED, TO BLANK.
         ELIMINATE EXCESS BLANKS BETWEEN THE STUDENT'S LAST NAME
         AND THE ',' IN NAME.
                                                                      */
BLANK:   DO TEAM# = MIN# TO MAX#;
           DO SEMINAR# = 1 TO 4;
             DO NUM = 1 TO 10;
               IF GRAD.NAME(TEAM#,SEMINAR#,NUM) = '0'
                 THEN GRAD.NAME(TEAM#,SEMINAR#,NUM) = ' ';
                 ELSE DO;
                   STR = SUBSTR(GRAD.NAME(TEAM#,SEMINAR#,NUM),
                     1,INDEX(GRAD.NAME(TEAM#,SEMINAR#,NUM),' ')
                     - 1) || SUBSTR(GRAD.NAME(TEAM#,SEMINAR#,
                     NUM),INDEX(GRAD.NAME(TEAM#,SEMINAR#,NUM),
                     ','));
                   GRAD.NAME(TEAM#,SEMINAR#,NUM) = STR;
                   END;
           END BLANK;
         /*

         PRINT A LIST FOR EACH SEMINAR, ONE LIST PER PAGE, OF
         EXPECTED GRADUATES WITH INFORMATION ABOUT MISSING
         LEARNING ACTIVITIES.
                                                                    , */
         DO TEAM# = MIN# TO MAX#;
           IF BIT(TEAM#) = '1'B THEN GO TO OUTER1;
           DO SEMINAR# = 1 TO 4;
             IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO INNER1;
             /*
             PRINT THE HEADER INFORMATION FOR THE FIRST PAGE OF
             PRINT.
                                                                      */
             PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
             PUT FILE(SYSPRINT) EDIT('CHECK')(SKIP,A);
             PUT FILE(SYSPRINT) EDIT('GRADUATION LIST:','YR = ',
               TEST_DATE.YR,'QTR = ',TEST_DATE.QTR)(SKIP,A,X(2),
               A,F(2),X(2),A,F(1));
             PUT FILE(SYSPRINT) EDIT('OUTPUT DESCRIPTION: ',
               'MISSING LEARNING ACTIVITIES BY AREA AND ',
               'TOTAL NUMBER OF MISSING ACTIVITIES','TEAM = ',
               TEAM#,'SEMINAR = ',FACULTY(FACULTY_NO(TEAM#,
               SEMINAR#)))(SKIP,3 A,SKIP,A,F(1),X(2),2 A);
```

```
                  PUT FILE(SYSPRINT) EDIT(('AREA',I DO I = 1 TO 11),
                    'TOTAL')(SKIP(5),COL(26),12 (A,X(1),F(2),X(1)));
                  PUT FILE(SYSPRINT) EDIT(('_______' DO I = 1 TO 11),
                    '_____')(SKIP(0),X(25),12 (A,X(1)));
                  /*

                  PRINT IN ALPHABETICAL ORDER BY LAST NAME THE NAME,
                  MISSING LEARNING ACTIVITY NUMBERS, AND TOTAL NUMBER
                  OF MISSING LEARNING ACTIVITIES FOR EACH EXPECTED
                  GRADUATE.
                                                                     */
                  DO NUM = 1 TO 10;
                     IF GRAD.NAME(TEAM#,SEMINAR#,NUM) = ' ' THEN
                       GO TO INNER1;
                     PUT FILE(SYSPRINT) EDIT(GRAD.NAME(TEAM#,SEMINAR#,
                       NUM))(SKIP(3),A);
                     DO I = 1 TO 5;
                       IF SUM(GRAD.ACTIVITY(TEAM#,SEMINAR#,NUM,*,I)) =
                         0 THEN GO TO F_END;
                       PUT FILE(SYSPRINT) EDIT(GRAD.ACTIVITY(TEAM#,
                         SEMINAR#,NUM,*,I))(COL(29),11 (F(2),X(6)));
                       END;
P_END:               PUT FILE(SYSPRINT) EDIT(GRAD.TOTAL(TEAM#,
                       SEMINAR#,NUM))(COL(115),F(3));
                     END;
INNER1:           END;
OUTER1:         END;
              /*

              PRINT A LIST OF EXPECTED GRADUATES FOR EACH TEAM.
              PRINT ONE TEAM PER PAGE.
                                                                     */
              DO TEAM# = MIN# TO MAX#;
                 IF BIT(TEAM#) = '1'B THEN GO TO OUTER;
                 /*

                 PRINT THE HEADER INFORMATION FOR EACH PAGE OF PRINT.
                                                                     */
                 PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
                 PUT FILE(SYSPRINT) EDIT('CHECK')(SKIP,A);
                 PUT FILE(SYSPRINT) EDIT('GRADUATION LIST:','YR = ',
                   TEST_DATE.YR,'QTR = ',TEST_DATE.QTR)(SKIP,A,X(2),A,
                   F(2),X(2),A,F(1));
                 PUT FILE(SYSPRINT) EDIT('OUTPUT DESCRIPTION: ',
                   'ALPHABETICAL LIST OF EXPECTED GRADUATES IN A TEAM '
                   ,'BY SEMINAR.  STUDENTS WITH MORE THAN 5 LEARNING ',
                   'ACTIVITIES TO COMPLETE ARE MARKED BY AN *','TEAM',
                   TEAM#)(SKIP,2 A,SKIP,X(2),2 A,SKIP,A,X(2),F(1));
                 PUT SKIP(10);
                 NONBLANK = 0;
                 /*

                 PROCESS THE DIFFERENT SEMINARS WITHIN THE TEAM.
                                                                     */
                 DO SEMINAR# = 1 TO 4;
                   IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO INNER;
                   NONBLANK = NONBLANK + 1;
                   /*

                   PRINT THE SEMINAR LEADER'S NAME.
                                                                     */
                   PUT FILE(SYSPRINT) EDIT(FACULTY(FACULTY_NO(TEAM#,
                     SEMINAR#)))(X(11),A);
INNER:             END;
                 PUT FILE(SYSPRINT) EDIT(('_______________'  DO I = 1
                   TO NONBLANK))(SKIP(0),(NONBLANK)(X(11),A));
                 PUT SKIP(2);
                 /*

                 PRINT THE NAMES OF EXPECTED GRADUATES BY SEMINAR IN
                 ALPHABETICAL ORDER BY LAST NAME.  MARK THOSE STUDENTS
                 WITH MORE THAN FIVE LEARNING ACTIVITIES TO COMPLETE
                 WITH AN '*'.
                                                                     */
                 DO NUM = 1 TO 10;
                     PUT SKIP;
                     KOUNT = 11;
                     DO SEMINAR# = 1 TO 4;
                     IF FACULTY_NO(TEAM#,SEMINAR#) = 0 |
                       GRAD.NAME(TEAM#,SEMINAR#,NUM) = ' ' THEN GO TO
                       INNER2;
                     IF GRAD.TOTAL(TEAM#,SEMINAR#,NUM) > 5
                       THEN PUT FILE(SYSPRINT) EDIT('*',GRAD.NAME
                         (TEAM#,SEMINAR#,NUM))(X(KOUNT - 1),2 A);
```

```
                ELSE PUT FILE(SYSPRINT) EDIT(GRAD.NAME(TEAM#,
                  SEMINAR#,NUM))(X(KOUNT),A);
              KOUNT = 1;
INNER2:       END;
            END;
OUTER:    END;
        END;
FINISH: END;
```

```
REC:     PROCEDURE OPTIONS(MAIN);
         /*

         PROGRAM:          FILE

         OUTPUT
         DESCRIPTION:      ASCENDING LIST OF THE STUDENT RECORDS IN
                           THE MAIN FILE BY SOCIAL SECURITY
                           NUMBER.

         ALGORITHM:        READ THE INITIAL DATA.  PROCESS THE FILE
                           SEQUENTIALLY BY READING A RECORD AND
                           PRINTING SPECIFIC INFORMATION FROM THE
                           RECORD.  ON ENCOUNTERING AN END OF FILE
                           TERMINATE THE EXECUTION OF THE PROGRAM.

         **********************************************************

         NOTE              THE FOLLOWING DECLARE STATEMENT(S)
                           INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                           ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                                1.   THE STATEMENT LABELED ONE WHICH
                                     CONTAINS THE STRUCTURE PERSONAL.

         **********************************************************
                                                                  */
         /*

         READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
         USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
         FACULTY_NO.
                                                                  */
         GET FILE(SYSIN) EDIT(MIN,MAX)(COL(1),2 (F(1),X(4)));
         BEGIN;
         DECLARE
           /*
           DICTIONARY OF FACULTY MEMBER'S NAMES ASSOCIATED WITH
           THE CODES RECORDED IN PERSONAL.                        */
           FACULTY(20) CHARACTER(14) INITIAL('BERNARD','CURRAN',
              'DETURE','FEY','GOODMAN','HILLIARD','JESTER',
              'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
              'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
              'TISON','WILMOT'),
           /*
           FACULTY NUMBERS OF THE SEMINAR LEADERS BY TEAM.        */
           FACULTY_NO(MIN:MAX,4) FIXED BINARY,
           CUR_DATE CHARACTER(8),
           OLD FILE RECORD SEQUENTIAL;
         /*

         ON ENCOUNTERING THE END OF THE MAIN FILE TRANSFER TO
         THE LABEL FINISH.
                                                                  */
         ON ENDFILE(OLD) GO TO FINISH;
         /*

         ON ENCOUNTERING THE END OF A PAGE OF PRINT, PRINT THE
         HEADER INFORMATION.
                                                                  */
         ON ENDPAGE(SYSIN) BEGIN;
           PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
           PUT FILE(SYSPRINT) EDIT('FILE',
              'OUTPUT DESCRIPTION: ASCENDING LIST OF THE STUDENT',
              ' RECORDS IN THE MAIN FILE BY SOCIAL SECURITY ',
              'NUMBER')(SKIP,A,SKIP,3 A);
           PUT FILE(SYSPRINT) EDIT('SOCIAL SECURITY NO','NAME',
              'TEAM','SEMINAR','___________________','____','____',
              '_______')(SKIP(6),A,COL(25),A,COL(60),A,COL(70),A,
              SKIP(0),A,COL(25),A,COL(60),A,COL(70),A);
           END;
         /*

         ON ENCOUNTERING AN ILLEGAL SUBSCRIPT TRANSFER TO LOOP TO
         PROCESS THE NEXT RECORD.
                                                                  */
         ON SUBSCRIPTRANGE GO TO LOOP;
         /*

         READ THE CURRENT DATE.
                                                                  */
         GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
         /*
```

```
        READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
        TEAM REAC THE TEAM NUMBER, THE NUMBER OF SEMINARS TO BE
        PROCESSCD, AND THE SEMINAR NUMBER AND FACULTY NUMBER
        OF EACH SEMINAR TO BE PROCESSED.
                                                                   */
        GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
        DO I = 1 TO NO_CARDS;
          GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
            TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
            X(4)),4 (F(1),X(4),F(2),X(3)));
            END;
        /*

        PRINT THE HEADER INFORMATION FOR THE FIRST PAGE OF
        PRINT.
                                                                   */
        PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
        PUT FILE(SYSPRINT) EDIT('FILE',
          'OUTPUT DESCRIPTION: ASCENDING LIST OF THE STUDENT',
          ' RECORDS IN THE MAIN FILE BY SOCIAL SECURITY ',
          'NUMBER')(SKIP,A,SKIP,3 A);
        PUT FILE(SYSPRINT) EDIT('SOCIAL SECURITY NO','NAME',
          'TEAM','SEMINAR','__________________','____','____',
          '_______')(SKIP(6),A,COL(25),A,COL(60),A,COL(70),A,
          SKIP(0),A,COL(25),A,COL(60),A,COL(70),A);
        /*

        PROCESS THE FILE SEQUENTIALLY.
                                                                   */
LCCF:   READ FILE(OLD) INTO(PERSCNAL);
        PUT FILE(SYSPRINT) EDIT(SCC_SEC_NO,SUBSTR(NAME.LAST,1,
          INDEX(NAME.LAST,' ')- 1) || ',' || NAME.FIRST,TEAM,
          FACULTY(FACULTY_NO(TEAM,SEMINAR)))(SKIP,A,COL(25),A,
          COL(60),F(1),COL(70),A);
        GO TC LCCP;
        END;
FINISH: END;
```

```
GRADS:  PROCEDURE OPTIONS(MAIN);
        /*
        PROGRAM:          GRADS

        OUTPUT
        DESCRIPTION:      ASCENDING LIST BY SOCIAL SECURITY
                          NUMBER OF STUDENTS WHOSE RECORDS WERE
                          PURGED FROM THE MAIN FILE AND DESTROYED
                          UNLESS OTHERWISE INDICATED.

        ALGORITHM:        READ THE INITIAL DATA.  PROCESS THE MAIN
                          FILE BY STORING SELECTED DATA FROM THE
                          PURGED RECORDS IN THE DELETE FILE AND
                          CREATING A NEW MAIN FILE.  PRINT AN
                          APPROPRIATE MESSAGE FOR EACH RECORD TO BE
                          PURGED.

        ***********************************************************

        NOTE              THE FOLLOWING DECLARE STATEMENT(S)
                          INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                          ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                               1.  THE STATEMENT LABELED ONE WHICH
                                   CONTAINS THE STRUCTURE PERSONAL.

        ***********************************************************

                                                                  */
        DECLARE
          /*
          BIT WHEN SET TO '1'B INDICATES THAT THE END OF FILE
          HAS BEEN ENCOUNTERED ON AN INPUT DEVICE.                */
          BIT BIT(1),
          /*
          CONTAINS SELECTED INFORMATION TO BE RETAINED ON ALL
          CEP GRADUATES.                                          */
          1 GRADUATE,
            2 SOC_SEC_NO CHARACTER(9),
            2 NAME,
              3 LAST CHARACTER(14),
              3 FIRST CHARACTER(10),
              3 MIDDLE CHARACTER(14),
            2 PERM_ADDR,
              3 STREET CHARACTER(20),
              3 CITY CHARACTER(20),
              3 STATE CHARACTER(2),
              3 ZIP CHARACTER(5),
          CUR_DATE CHARACTER(8),
          /*
          EXPECTED GRADUATION DATE TO BE CHECKED.                 */
          1 TEST_DATE,
              3 YR FIXED BINARY,
              3 QTR FIXED BINARY,
          (OLD,NEW,DELETE) FILE RECORD SEQUENTIAL,
          SS_NO CHARACTER(9);
        BIT = '0'B;
        /*

        ON ENCOUNTERING AN END OF FILE ON SYSIN, WHICH INDICATES
        THAT THERE ARE NO MORE RECORDS TO BE PURGED FROM THE
        MAIN FILE, SET BIT TO A '1'B.  COPY THE REMAINING
        RECORDS ONTO THE NEW MAIN FILE.
                                                                  */
        ON ENDFILE(SYSIN) BEGIN;
          IF BIT = '1'B THEN GO TO FINISH;
          BIT = '1'B;
OTHERS:   READ FILE(OLD) INTO(PERSONAL);
          WRITE FILE(NEW) FROM(PERSONAL);
          GO TO OTHERS;
          END;
        /*

        ON ENCOUNTERING AN END OF FILE ON OLD, WHICH INDICATES
        THAT NO MORE RECORDS EXIST IN THE MAIN FILE, SET BIT TO
        A '1'B.  PRINT THE REMAINING SOCIAL SECURITY NUMBERS TO
        BE PURGED WITH THE MESSAGE 'MISSING.'
                                                                  */
        ON ENDFILE(OLD) BEGIN;
          IF BIT = '1'B THEN GO TO FINISH;
          BIT = '1'B;
MESSAGE:  PUT FILE(SYSPRINT) EDIT('MISSING',SS_NO)(SKIP,A,X(12),
            A);
          GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
```

```
            GO TO MESSAGE;
            END;
          /*

          ON ENCOUNTERING THE END OF A PAGE OF PRINT, PRINT THE
          HEADER INFORMATION.
                                                                 */
          ON ENDPAGE(SYSPRINT) BEGIN;
            PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
            PUT FILE(SYSPRINT) EDIT('GRADS',
              'OUTPUT DESCRIPTION: ASCENDING LIST OF GRADUATES ',
              'WHOSE RECORDS WERE PURGED',' FROM THE MAIN FILE ',
              'AND DESTROYED UNLESS OTHERWISE INDICATED')(SKIP,A,
              SKIP,2 A,COL(5),2 A);
            PUT FILE(SYSPRINT) EDIT('ERROR MESSAGE','S.S.NUMBER',
              'NAME','ADDRESS','_____________________','_______________','____'
              ,'________')(SKIP(6),A,COL(20),A,COL(40),A,COL(65),A,
              SKIP(0),A,COL(20),A,COL(40),A,COL(65),A);
            END;
          /*

          READ THE CURRENT DATE AND THE GRADUATION TEST DATE.
                                                                 */
          GET FILE(SYSIN) EDIT(CUR_DATE,TEST_DATE)(COL(1),A(8),
            SKIP,F(2),X(3),F(1));
          /*

          PRINT THE HEADER INFORMATION FOR THE FIRST PAGE OF
          PRINT.
                                                                 */
          PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
          PUT FILE(SYSPRINT) EDIT('GRADS',
            'OUTPUT DESCRIPTION: ASCENDING LIST OF GRADUATES ',
            'WHOSE RECORDS WERE PURGED',' FROM THE MAIN FILE ',
            'AND DESTROYED UNLESS OTHERWISE INDICATED')(SKIP,A,
            SKIP,2 A,COL(5),2 A);
          PUT FILE(SYSPRINT) EDIT('ERROR MESSAGE','S.S.NUMBER',
            'NAME','ADDRESS','_____________________','_______________','____'
            ,'________')(SKIP(6),A,COL(20),A,COL(40),A,COL(65),A,
            SKIP(0),A,COL(20),A,COL(40),A,COL(65),A);
          /*

          READ THE SOCIAL SECURITY NUMBER OF A RECORD TO BE
          PURGED FROM THE MAIN FILE.
                                                                 */
NEXT:     GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
          /*

          READ A RECORD FORM THE MAIN FILE.
                                                                 */
LOOP:     READ FILE(OLD) INTO(PERSONAL);
CHECK:
          /*

          IF THE RECORD TO BE PURGED HAS NOT YET BEEN FOUND IN THE
          MAIN FILE, COPY THE CURRENT RECORD ONTO THE NEW MAIN
          FILE.
                                                                 */
          IF PERSONAL.SOC_SEC_NO < SS_NO THEN DO;
            WRITE FILE(NEW) FROM(PERSONAL);
            GO TO LOOP;
            END;
          /*

          IF THE RECORD TO BE PURGED DOES NOT EXIST IN THE MAIN
          FILE, PRINT THE SOCIAL SECURITY NUMBER WITH THE MESSAGE
          'MISSING.'  READ THE SOCIAL SECURITY NUMBER OF THE NEXT
          RECORD TO BE PURGED.
                                                                 */
          IF PERSONAL.SOC_SEC_NO > SS_NO THEN DO;
            WRITE FILE(NEW) FROM(PERSONAL);
            PUT FILE(SYSPRINT) EDIT('MISSING',SS_NO)
              (SKIP,A,X(12),A);
            GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
            GO TO CHECK;
            END;
          /*

          THE RECORD TO BE PURGED HAS BEEN FOUND.  CHECK THE
          GRADUATION DATE TO VERIFY THAT THE RECORD SHOULD BE
          PURGED AT THIS TIME.  IF THERE IS AN INCONSISTENCY IN
          DATES, PRINT THE SOCIAL SECURITY NUMBER WITH THE MESSAGE
          'INCORRECT DATE' AND OBTAIN A NEW RECORD.  OTHERWISE,
          PRINT THE SOCIAL SECURITY NUMBER, NAME AND PERMANENT
```

```
         ADDRESS FROM THIS RECORD.  ASSIGN TO GRADUATE SELECTED
         DATA FROM PERSONAL AND COPY GRADUATE ONTO THE DELETE
         FILE,
                                                                   */
         IF PERSONAL.EXP_GRAD_DATE.YR ¬= TEST_DATE.YR &
           PERSONAL.EXP_GRAD_DATE.QTR ¬= TEST_DATE.QTR THEN DO;
           WRITE FILE(NEW) FROM(PERSONAL);
           PUT FILE(SYSPRINT) EDIT('INCORRECT DATE',
             PERSONAL.SOC_SEC_NO)(SKIP,A,X(5),A);
           GO TO NEXT;
           END;
         GRACUATE = PERSCNAL, BY NAME;
         PUT FILE(SYSPRINT) EDIT(GRADUATE.SOC_SEC_NO,
           SUBSTR(GRADUATE.LAST,1,INDEX(GRADUATE.LAST,' ')-1) ||
           ',' || GRADUATE.FIRST,GRADUATE.PERM_ADDR)(COL(20),A,
           X(9),A,COL(65),4 (A,X(2)));
         WRITE FILE(DELETE) FROM(GRADUATE);
         GO TO NEXT;
FINISH: END;
```

```
A_LIST:    PROCEDURE OPTIONS(MAIN);
         /*

         PROGRAM:         LOCAL

         OUTPUT
         ALGORITHM:       ALPHABETICAL LIST OF TEAM MEMBER'S
                          ADDRESSES BY SEMINAR.

         ALGORITHM:       READ THE INITIAL DATA.  PROCESS THE MAIN
                          FILE BY STORING IN THE STRUCTURE ROSTER
                          THE NAME AND LOCAL ADDRESS OF STUDENTS IN
                          THE SEMINARS BEING PROCESSED.  ON
                          ENCOUNTERING THE END OF THE MAIN FILE
                          SORT THE STRUCTURE ROSTER ALPHABETICALLY
                          BY THE NAMES.  ELIMINATE EXCESS BLANKS
                          FROM THE NAMES IN THE STRUCTURE ROSTER.
                          PRINT THE ADDRESS LISTS BY SEMINAR.

         ********************************************************

         NOTE             THE FOLLOWING DECLARE STATEMENT(S)
                          INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                          ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                               1.  THE STATEMENT LABELED ONE WHICH
                                   CONTAINS THE STRUCTURE PERSONAL.

         ********************************************************
                                                                      */
         /*


         READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
         USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
         ROSTER, FACULTY_NO AND VALUE.
                                                                      */
         GET FILE(SYSIN) EDIT(MIN,MAX)(COL(1),2 (F(1),X(4)));
         BEGIN;
         DECLARE
           /*
           INDICATOR OF THE PRINT STATUS FOR THIS TEAM.               */
           BIT(MIN:MAX) BIT(1),
           /*
           DICTIONARY OF FACULTY MEMBER'S NAMES ASSOCIATED WITH
           THE CODES RECORDED IN PERSONAL.                            */
           FACULTY(20) CHARACTER(14) INITIAL('BERNARD','CURRAN',
              'DETURE','FEY','GOODMAN','HILLIARD','JESTER',
              'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
              'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
              'TISON','WILMOT'),
           /*
           FACULTY NUMBERS OF THE SEMINAR LEADERS BY TEAM.            */
           FACULTY_NO(MIN:MAX,4) FIXED BINARY,
           /*
           NAME AND LOCAL ADDRESS OF STUDENTS IN EACH TEAM BY
           SEMINAR.                                                   */
           1 ROSTER(MIN:MAX,4,35),
             2 NAME CHARACTER(25),
             2 ADDRESS,
               3 STREET CHARACTER(20),
               3 CITY CHARACTER(20),
             2 PHONE CHARACTER(7),
           /*
           COUNTER FOR THE NUMBER OF STUDENTS IN EACH SEMINAR BY
           TEAM.                                                      */
           VALUE(MIN:MAX,4) FIXED BINARY,
           CUR_DATE CHARACTER(8),
           OLD FILE RECORD SEQUENTIAL,
           SEMINAR# FIXED BINARY,
           STR CHARACTER(25),
           TEAM# FIXED BINARY,
           1 TEMP,
             2 NAME CHARACTER(25),
             2 ADDRESS,
               3 STREET CHARACTER(20),
               3 CITY CHARACTER(20),
             2 PHONE CHARACTER(7);
         /*

         INITIALIZE VALUE, ROSTER AND FACULTY NUMBER.
                                                                      */
         VALUE = 0;
         ROSTER = '0';
         FACULTY_NO = 0;
         /*

         ON ENCOUNTERING AN ILLEGAL SUBSCRIPT IN A STUDENT'S
         RECORD, PRINT HIS SOCIAL SECURITY NUMBER, TEAM NUMBER,
         SEMINAR NUMBER AND THE TOTAL NUMBER OF STUDENTS IN THIS
         SEMINAR.
                                                                 */
         ON SUBSCRIPTRANGE BEGIN;
           PUT FILE(SYSPRINT) EDIT('SUBSCRIPT OUTSIDE LEGAL',
             ' BOUNDS','SOC_SEC_NO = ',SOC_SEC_NO,'TEAM = ',TEAM,
             'SEMINAR = ',SEMINAR,'VALUE = ',VALUE(TEAM,SEMINAR))
             (2 A,SKIP(3),A,A,X(5),3 (A,F(2),X(5)));
           GO TO FINISH;
           END;
         /*

         ON ENCOUNTERING THE END OF THE MAIN FILE TRANSFER TO
         THE LABEL SORT.
                                                                 */
         ON ENDFILE(OLD) GO TO SORT;
         OPEN FILE(SYSPRINT) LINESIZE(130);
         /*

         READ THE CURRENT DATE.
                                                                 */
         GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
         /*

         READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
         TEAM READ THE TEAM NUMBER, THE NUMBER OF SEMINARS TO BE
         PROCESSED, AND THE SEMINAR NUMBER AND FACULTY NUMBER
         OF EACH SEMINAR TO BE PROCESSED.
                                                                 */
         GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
         DO I = 1 TO NO_CARDS;
           GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
             TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
             X(4)),4 (F(1),X(4),F(2),X(3)));
             END;
         /*

         PROCESS THE FILE SEQUENTIALLY.  READ A STUDENT'S RECORD.
         IF THE RECORD IS NOT TO BE PRINTED, TRANSFER TO LOOP TO
         GET A NEW RECORD.  INCREMENT THE CORRECT COUNTER IN
         VALUE BY ONE.  STORE THE STUDENT'S NAME, LAST AND FIRST
         NAMES ONLY, AND LOCAL ADDRESS IN THE CORRECT POSITION
         IN THE STRUCTURE ROSTER.
                                                                 */
LOOP:    READ FILE(OLD) INTO(PERSONAL);
         IF TEAM < MIN | TEAM > MAX THEN GO TO LOOP;
         IF FACULTY_NO(TEAM,SEMINAR) = 0 THEN GO TO LOOP;
         VALUE(TEAM,SEMINAR) = VALUE(TEAM,SEMINAR) + 1;
         ROSTER.NAME(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) =
           PERSONAL.NAME.LAST || ',' || PERSONAL.NAME.FIRST;
         ROSTER.ADDRESS.STREET(TEAM,SEMINAR,VALUE(TEAM,SEMINAR))
           = PERSONAL.LOCAL_ADDR.STREET;
         ROSTER.ADDRESS.CITY(TEAM,SEMINAR,VALUE(TEAM,SEMINAR))
           = PERSONAL.LOCAL_ADDR.CITY;
         ROSTER.PHONE(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) =
           PERSONAL.LOCAL_PHONE;
         GO TO LOOP;
         /*

         SORT THE SEMINARS IN THE STRUCTURE ROSTER ALPHABETICALLY
         BY NAME.
                                                                 */
SORT:    DO TEAM# = MIN TO MAX;
           BIT(TEAM#) = FACULTY_NO(TEAM#,1) = 0 &
             FACULTY_NO(TEAM#,2) = 0 & FACULTY_NO(TEAM#,3) = 0 &
             FACULTY_NO(TEAM#,4) = 0;
           IF BIT(TEAM#) = '1'B THEN GO TO T_END;
           DO SEMINAR# = 1 TO 4;
             IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO S_END;
             DO I = 1 TO 34;
               DO J = I+1 TO 35;
                 IF ROSTER.NAME(TEAM#,SEMINAR#,I) >
                   ROSTER.NAME(TEAM#,SEMINAR#,J) THEN DO;
                   TEMP = ROSTER(TEAM#,SEMINAR#,I);
                   ROSTER(TEAM#,SEMINAR#,I) = ROSTER(TEAM#,
                     SEMINAR#,J);
                   ROSTER(TEAM#,SEMINAR#,J) = TEMP;
                   END;
               END;
```

```
            END;
S_END:    END;
T_END:  END;
        /*

        PROCESS THE NAMES IN THE STRUCTURE ROSTER.  CHANGE ALL
        DUMMY NAMES, NAMES WHICH HAD BEEN INITIALIZED TO '0' FOR
        SORTING AND HAVE NOT BEEN CHANGED, TO BLANK.  ELIMINATE
        EXCESS BLANKS BETWEEN THE STUDENT'S LAST NAME IN NAME
        AND THE ',' IN THE REMAINDER OF NAME.
                                                              */
BLANK:  DO TEAM# = MIN TO MAX;
          DO SEMINAR# = 1 TO 4;
            DO NUM = 1 TO 36;
              IF ROSTER.NAME(TEAM#,SEMINAR#,NUM) = '0'
                THEN ROSTER(TEAM#,SEMINAR#,NUM) = ' ';
                ELSE DO;
                  STR = SUBSTR(ROSTER.NAME(TEAM#,SEMINAR#,NUM),
                    1,INDEX(ROSTER.NAME(TEAM#,SEMINAR#,NUM),' ')
                    - 1) || SUBSTR(ROSTER.NAME(TEAM#,SEMINAR#,
                    NUM),INDEX(ROSTER.NAME(TEAM#,SEMINAR#,NUM),
                    ','));
                  ROSTER.NAME(TEAM#,SEMINAR#,NUM) = STR;
                  END;
          END BLANK;
        /*

        PRINT AN ADDRESS LIST FOR EACH SEMINAR. PRINT ONE
        SEMINAR PER PAGE.
                                                              */
        DO TEAM# = MIN TO MAX;
          IF BIT(TEAM#) = 'I'B THEN GO TO OUTER;
          DO SEMINAR# = 1 TO 4;
            IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO INNER;
            /*

            PRINT THE HEADER INFORMATION FOR EACH PAGE OF PRINT.
                                                              */
            PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
            PUT FILE(SYSPRINT) EDIT('LOCAL',
              'OUTPUT DESCRIPTION: ALPHABETICAL LIST OF TEAM ',
              'MEMBERS ADDRESSES BY SEMINAR','TEAM',TEAM#)(SKIP,
              A,SKIP,2 A,SKIP,A,X(2),F(2));
            PUT SKIP(10);
            /*

            PRINT THE SEMINAR LEADER'S NAME.
                                                              */
            PUT FILE(SYSPRINT) EDIT(FACULTY(FACULTY_NO(TEAM#,
              SEMINAR#)),'______________')(X(10),A,SKIP(3),X(10)
              ,A);
            /*

            PRINT IN ALPHABETICAL ORDER BY LAST NAME THE NAME
            AND LOCAL ADDRESS OF SEMINAR MEMBERS IN COLUMNS OF
            TWELVE STUDENTS PER COLUMN.
                                                              */
            DO NUM = 1 TO 12;
              PUT FILE(SYSPRINT) SKIP(2);
              PUT FILE(SYSPRINT) EDIT((ROSTER.NAME(TEAM#,
                SEMINAR#,I) DO I = NUM TO 36 BY 12),
                (ROSTER.ADDRESS.STREET(TEAM#,SEMINAR#,I) DO I =
                NUM TO 36 BY 12),(ROSTER.ADDRESS.CITY(TEAM#,
                SEMINAR#,I) DO I = NUM TO 35 BY 12),
                (ROSTER.PHONE(TEAM#,SEMINAR#,I) DO I = NUM TO 36
                BY 12))
                (COL(10),A,COL(50),A,COL(90),A);
              END;
INNER:      END;
OUTER:    END;
        END;
FINISH: END;
```

```
PRINT:  PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:         PRINT

        OUTPUT
        DESCRIPTION:     LISTING OF SPECIFIC DATA ON CERTAIN
                         STUDENTS AS REQUESTED BY FACULTY MEMBERS.

        ALGORITHM:       READ THE INITIAL DATA.  OBTAIN A FACULTY
                         REQUEST.  IF THE REQUEST TYPE IS EQUAL
                         TO ONE, PRINT THE REQUESTED INFORMATION
                         ON ALL MEMBERS OF THE SPECIFIED SEMINAR.
                         OTHERWISE, PRINT THE REQUESTED
                         INFORMATION ON THE SOCIAL SECURITY
                         NUMBERS FOLLOWING THE REQUEST UNTIL AN
                         '*', WHICH INDICATES THE END OF THIS
                         REQUEST, IS READ.  CONTINUE TO PROCESS
                         THE REMAINING REQUESTS IN THE SAME MANNER.

        ********************************************************

        NOTE             THE FOLLOWING DECLARE STATEMENT(S)
                         INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                         ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                              1.  THE STATEMENT LABELED ONE WHICH
                                  CONTAINS THE STRUCTURE PERSONAL.
                              2.  THE STATEMENT LABELED TWO WHICH
                                  CONTAINS THE DICTIONARIES.

        ********************************************************

                                                                       */
        DECLARE
          /*
          AN ENTRY POINT, FOR INSTANCE ENTRY(I), INDICATES THE
          NUMBER OF THE POSITION PRECEEDING THE BEGINNING OF
          THIS SUBSTANTIVE AREA IN LEARNING_A.                        */
          ENTRY(12) FIXED BINARY INITIAL(0,6,19,31,41,50,56,66,
            82,87,95,99),
          /*
          DICTIONARY OF THE FACULTY MEMBER'S NAMES ASSOCIATED
          WITH THE CODES RECORDED IN PERSONAL.                        */
          FACULTY(26) CHARACTER(14) INITIAL('BERNARD','CURRAN',
            'DETURE','FEY','GOLDMAN','HILLIARD','JESTER',
            'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
            'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
            'TISON','WILMOT','KRALL','CREEDON','LEVINE',
            'HODGES','GRIMM','DORMAN'),
          /*
          FLAG WHEN SET TO A '1'B INDICATES THAT AN END OF FILE
          HAS BEEN ENCOUNTERED ON AN INPUT DEVICE.                    */
          FLAG BIT(1) INITIAL('0'B),
          /*
          CONTAINS INFORMATION WHICH INDICATES THE PRINT STATUS
          AND BOUNDS OF EXECUTION FOR EACH REQUEST.                   */
          1 REQUEST(8),
            2 CODE FIXED BINARY,
            2 INDX,
              3 NUM1 FIXED BINARY,
              3 NUM2 FIXED BINARY,
          CUR_DATE CHARACTER(8),
          FACULTY# FIXED BINARY,
          OLD FILE RECORD SEQUENTIAL,
          SS_NO CHARACTER(9),
          SPACE FIXED BINARY;
        OPEN FILE(SYSPRINT) LINESIZE(130);
        /*

        ON ENCOUNTERING AN END OF FILE ON OLD, REWIND THE TAPE.
        IF FLAG IS SET TO A '1'B, READ THE NEXT SOCIAL SECURITY
        NUMBER.  IF SS_NO IS AN '*' RESET FLAG TO A '0'B AND
        OBTAIN A NEW REQUEST.  OTHERWISE, CONTINUE PROCESSING
        THE SOCIAL SECURITY NUMBERS.
                                                                       */
        ON ENDFILE(OLD) BEGIN;
          CLOSE FILE(OLD);
          OPEN FILE(OLD);
          IF FLAG = '1'B THEN DO;
            GET FILE(SYSIN) EDIT(SS_NO)(COL(1),A(9));
            IF SS_NO = '*' THEN DO;
              FLAG = '0'B;
              GO TO NEXT;
              END;
```

```
               ELSE GO TO LOOP;
             END;
           GO TO NEXT;
           END;
         /*

         ON ENCOUNTERING A SUBSCRIPTRANGE ERROR, PRINT AN ERROR
         MESSAGE AND TRANSFER TO PROCESS THE NEXT RECORD FOR
         THIS REQUEST.
                                                                    */
         ON SUBSCRIPTRANGE BEGIN;
           PUT FILE(SYSPRINT) EDIT(SOC_SEC_NO,'SUBSCRIPT OUTSIDE'
              ,' LEGAL BOUNDS')(A,COL(32),2 A);
           IF REQUEST_TYPE ¬= 1 THEN GO TO GROUPS;
             ELSE GO TO SINGLE;
           END;
         /*

         ON ENCOUNTERING THE END OF THE REQUESTS TRANSFER TO
         FINISH.
                                                                    */
         ON ENDFILE(SYSIN) GO TO FINISH;
         /*

         ON ENCOUNTERING AN END OF A PAGE, PRINT THE HEADER
         INFORMATION.
                                                                    */
         ON ENDPAGE(SYSPRINT) BEGIN;
           PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
           PUT FILE(SYSPRINT) EDIT('PRINT','OUTPUT DESCRIPTION: '
              ,'LISTING OF SPECIFIC DATA ON CERTAIN STUDENTS AS ',
              'REQUESTED BY FACULTY MEMBERS','REQUEST FOR ',
              FACULTY(FACULTY#))(SKIP,A,SKIP,3 A,SKIP,2 A);
           PUT FILE(SYSPRINT) EDIT('S.S. NUMBER AND NAME','ITEM',
              'CONTENTS','____________________','____','________')
              (SKIP(6),2 (COL(1),A,X(12),A,X(I1),A,SKIP(0)));
           PUT FILE(SYSPRINT) SKIP(2);
           END;
         /*

         READ THE CURRENT DATE.
                                                                    */
         GET FILE(SYSIN) EDIT(CUR_DATE)(COL(I),A(8));
         /*

         READ THE INITIAL DATA FOR THE NEXT FACULTY REQUEST.
                                                                    */
NEXT:    GET FILE(SYSIN) EDIT(FACULTY#,REQUEST_TYPE,TEAM#,
           SEMINAR#)(COL(1),F(2),X(3),3 (F(1),X(4)));
         GET FILE(SYSIN) EDIT(REQUEST)(COL(I),5 (F(1),X(4),2 (F
           (2),X(3))));
         REQ_COURSE_GRADE = ' ';
         /*

         IF THE REQUEST TYPE IS ONE, PROCESSING IS TO BE DONE
         BY SEMINARS.
                                                                    */
         IF REQUEST_TYPE = I THEN DO;
           /*

           READ A RECORD FROM THE MAIN FILE.
                                                                    */
GROUPS:    READ FILE(OLD) INTO(PERSONAL);
           IF PERSONAL.TEAM = TEAM# & PERSONAL.SEMINAR = SEMINAR#
             THEN DO;
             /*

             PRINT THE HEADER INFORMATION.
                                                                    */
             PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
             PUT FILE(SYSPRINT) EDIT('PRINT','OUTPUT DESCRIPTION'
               ,': LISTING OF SPECIFIC DATA ON CERTAIN STUDENTS '
               ,'AS REQUESTED BY FACULTY MEMBERS','REQUEST FOR ',
               FACULTY(FACULTY#))(SKIP,A,SKIP,3 A,SKIP,2 A);
             PUT FILE(SYSPRINT) EDIT('S.S. NUMBER AND NAME',
               'ITEM','CONTENTS','____________________','____',
               '________')(SKIP(6),2 (COL(I),A,X(12),A,X(II),A,
               SKIP(0)));
             /*

             PRINT THE STUDENT'S SOCIAL SECURITY NUMBER AND NAME
             .
                                                                    */
             PUT FILE(SYSPRINT) EDIT(SOC_SEC_NO,SUBSTR(
              PERSONAL.LAST,1,INDEX(PERSONAL.LAST,' ')-1) || ','
              || PERSONAL.FIRST)(SKIP(2),A,COL(6),A);
            SPACE = 0;
            /*

            CHECK EACH REQUEST CODE.  IF THE CODE IS NOT EQUAL
            TO ZERO, CALL THE CORRECT PRINT SUBROUTINE WITH THE
            CORRECT ARGUMENTS.
                                                                  */
            IF REQUEST.CODE(1) ¬= 0 THEN CALL SUB_1;
            IF REQUEST.CODE(2) ¬= 0 THEN CALL SUB_2(REQUEST.CODE
              (2),REQUEST.NUM1(2));
            IF REQUEST.CODE(3) ¬= 0 THEN CALL SUB_3(REQUEST.CODE
              (3),REQUEST.NUM1(3),REQUEST.NUM2(3));
            IF REQUEST.CODE(4) ¬= 0 THEN CALL SUB_4(REQUEST.CODE
              (4),REQUEST.NUM1(4),REQUEST.NUM2(4));
            IF REQUEST.CODE(5) ¬= 0 THEN CALL SUB_5(REQUEST.CODE
              (5),REQUEST.NUM1(5));
            IF REQUEST.CODE(6) ¬= 0 THEN CALL SUB_6(REQUEST.CODE
              (6),REQUEST.NUM1(6));
            IF REQUEST.CODE(7) ¬= 0 THEN CALL SUB_7(REQUEST.CODE
              (7),REQUEST.NUM1(7));
            IF REQUEST.CODE(8) ¬= 0 THEN CALL SUB_8;
           END;
          GO TO GROUPS;
          END;
          /*

          IF THE REQUEST TYPE IS ZERO, PROCESSING IS TO BE DONE
          BY INDIVIDUAL SOCIAL SECURITY NUMBERS.
                                                                  */
          ELSE DO;
          FLAG = '1'B;
          /*

          READ THE SOCIAL SECURITY NUMBER OF THE NEXT RECORD TO
          BE PRINTED.
                                                                  */
SINGLE:   GET FILE(SYSIN) EDIT(SS_NO)(COL(1),A(9));
          /*

          IF THE SOCIAL SECURITY NUMBER IS AN '*', WHICH
          INDICATES THE END OF A GROUP REQUEST, REWIND THE TAPE
          AND TRANSFER TO READ THE NEXT REQUEST.
                                                                  */
          IF SS_NO = '*' THEN DO;
            CLOSE FILE(OLD);
            OPEN FILE(OLD);
            GO TO NEXT;
            END;
          /*

          PRINT THE HEADER INFORMATION.
                                                                  */
          PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
          PUT FILE(SYSPRINT) EDIT('PRINT','OUTPUT DESCRIPTION: '
            ,'LISTING OF SPECIFIC DATA ON CERTAIN STUDENTS AS ',
            'REQUESTED BY FACULTY MEMBERS','REQUEST FOR ',
            FACULTY(FACULTY#))(SKIP,A,SKIP,3 A,SKIP,2 A);
          PUT FILE(SYSPRINT) EDIT('S.S. NUMBER AND NAME','ITEM',
            'CONTENTS','____________________','________','_________')
            (SKIP(6),2 (COL(1),A,X(12),A,X(11),A,SKIP(0)));
          /*

          READ A RECORD FROM THE MAIN FILE.
                                                                  */
LOOP:     READ FILE(OLD) INTO(PERSONAL);
          /*

          IF THE RECORD TO BE PRINTED HAS NOT YET BEEN FOUND IN
          THE MAIN FILE, OBTAIN A NEW RECORD.
                                                                  */
          IF PERSONAL.SOC_SEC_NO < SS_NO THEN GO TO LOOP;
          /*

          IF THE RECORD TO BE PRINTED DOES NOT EXIST IN THE MAIN
          FILE, PRINT THE SOCIAL SECURITY NUMBER WITH THE
          MESSAGE 'MISSING.'  READ THE NEXT SOCIAL SECURITY
          NUMBER.  IF IT IS AN '*', REWIND THE TAPE TO OBTAIN
          A NEW REQUEST.
                                                                  */
          IF PERSONAL.SOC_SEC_NO > SS_NO THEN DO;
            PUT FILE(SYSPRINT) EDIT(SS_NO,'*MISSING')(SKIP(4),
              A,X(20),A);
            GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
            IF SS_NO = '*' THEN DO;
              CLOSE FILE(OLD);
              OPEN FILE(OLD);
              GO TO NEXT;
              END;
            GO TO LOOP;
            END;
          /*

          PRINT THE STUDENT'S SOCIAL SECURITY NUMBER AND NAME.
                                                                */
          PUT FILE(SYSPRINT) EDIT(SOC_SEC_NO,SUBSTR(
            PERSONAL.LAST,1,INDEX(PERSONAL.LAST,' ')-1) || ','
            || PERSONAL.FIRST)(SKIP(2),A,COL(5),A);
          SPACE = 0;
          /*

          CHECK EACH REQUEST CODE.  IF THE CODE IS NOT EQUAL TO
          ZERO, CALL THE CORRECT PRINT SUBROUTINE WITH THE
          CORRECT ARGUMENTS.
                                                                */
          IF REQUEST.CODE(1) ¬= 0 THEN CALL SUB_1;
          IF REQUEST.CODE(2) ¬= 0 THEN CALL SUB_2(REQUEST.CODE
            (2),REQUEST.NUM1(2));
          IF REQUEST.CODE(3) ¬= 0 THEN CALL SUB_3(REQUEST.CODE
            (3),REQUEST.NUM1(3),REQUEST.NUM2(3));
          IF REQUEST.CODE(4) ¬= 0 THEN CALL SUB_4(REQUEST.CODE
            (4),REQUEST.NUM1(4),REQUEST.NUM2(4));
          IF REQUEST.CODE(5) ¬= 0 THEN CALL SUB_5(REQUEST.CODE
            (5),REQUEST.NUM1(5));
          IF REQUEST.CODE(6) ¬= 0 THEN CALL SUB_6(REQUEST.CODE
            (6),REQUEST.NUM1(6));
          IF REQUEST.CODE(7) ¬= 0 THEN CALL SUB_7(REQUEST.CODE
            (7),REQUEST.NUM1(7));
          IF REQUEST.CODE(8) ¬= 0 THEN CALL SUB_8;
          GO TO SINGLE;
NEW:      END;
        /*

        PRINT THE STUDENT'S MYERS BRIGGS TYPE.
                                                                */
SUB_1:  PROCEDURE;
        PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(1),
          MYERS_BRIGGS_DICT(PERSONAL.MYERS_BRIGGS))(SKIP(SPACE),
          COL(32),A,COL(46),A);
        SPACE = 3;
        END SUB_1;
        /*

        PRINT INFORMATION ABOUT THE STUDENT'S REQUIRES COURSES.
                                                                */
SUB_2:  PROCEDURE(CODE#,VALUE);
        DECLARE
          (CODE#,VALUE) FIXED BINARY;
        /*

        IF CODE# IS EQUAL TO TWO, PRINT INFORMATION ABOUT ALL
        ALL REQUIRED COURSES.  OTHERWISE, PRINT INFORMATION
        ABOUT ONE REQUIRED COURSE USING VALUE AS AN INDEX.
                                                                */
        IF CODE# = 2 THEN DO;
          INITIAL = 1;
          LIMIT = 9;
          END;
          ELSE INITIAL,LIMIT = VALUE;
        PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(2))(SKIP(SPACE),
          COL(32),A);
        /*

        PRINT EXPLANATORY INFROMATION ABOUT THE GRADE.
                                                                */
        PUT FILE(SYSPRINT) EDIT('GRADE = * INDICATES COURSE',
          ' REQUIREMENT HAS BEEN MET')(COL(46),2 A);
        DO NUM = INITIAL TO LIMIT;
          PUT FILE(SYSPRINT) EDIT(REQ_COURSE_DICT(NUM),
            'GRADE = ',REQ_COURSE_GRADE(NUM))(COL(46),A,X(4),
            2 A);
        SPACE = 3;
        END SUB_2;
        /*

        PRINT INFORMATION ABOUT THE STUDENT'S REGISTRATIONS.
                                                                */
```

```
SUB_3:   PROCEDURE(CODE#,VALUE_1,VALUE_2);
         DECLARE
           (CODE#,VALUE_1,VALUE_2) FIXED BINARY;
         /*

         IF CODE# IS EQUAL TO TWO, PRINT INFORMATION ABOUT ALL
         COMPLETED REGISTRATIONS.  OTHERWISE, PRINT INFORMATION
         ABOUT ONE REGISTRATION.  IF CODE# IS EQUAL TO ONE, USE
         VALUE_1 AS AN INDEX.  IF CODE# IS EQUAL TO THREE, FIND
         THE INDEX BY CHECKING VALUE_1 AND VALUE_2.
                                                                  */
         IF CODE# = 2 THEN DO;
           INITIAL = 1;
           LIMIT = 8;
           END;
           ELSE DO;
             INITIAL,LIMIT = VALUE_1;
             IF CODE# = 3 THEN DO;
               DO NUM = 1 TO 8;
                 IF VALUE_1 = REGISTRATION(NUM).YR &
                   VALUE_2 = REGISTRATION(NUM).QTR THEN GO TO
                   PROCESS;
                 END;
               GO TO LSUB_3;
               END;
             END;
PROCESS:PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(3))(SKIP(SPACE),
           COL(32),A);
         DO NUM = INITIAL TO LIMIT WHILE(REGISTRATION(NUM).YR ¬=
           0);
           PUT FILE(SYSPRINT) EDIT('YR = ',REGISTRATION(NUM).YR,
             'QTR = ',REGISTRATION(NUM).QTR)(COL(46),A,F(2),X(3),
             A,F(1));
           DO KOUNT = 1 TO 5 WHILE(COURSE_NO(NUM,KOUNT) ¬= ' ');
             PUT FILE(SYSPRINT) EDIT(COURSE_NO(NUM,KOUNT),
               'HRS = ',HRS(NUM,KOUNT),'GRADE = ',REGISTRATION
               (NUM).GRADE(KOUNT))(COL(66),A,X(4),A,F(2),X(2),2 A
               );
             END;
           END;
         SPACE = 3;
LSUB_3:  END SUB_3;
         /*

         PRINT ALL INFORMATION CONTAINED IN THE STRUCTURE
         PERSONAL ABOUT THE STUDENT'S LEARNING ACTIVITY OR
         ACTIVITIES.
                                                                  */
SUB_4:   PROCEDURE(CODE#,VALUE_1,VALUE_2);
         DECLARE
           GP(2) CHARACTER(10) INITIAL('GROUP','INDIVIDUAL'),
           SUB(2) CHARACTER(10) INITIAL('SUBSTITUTE','REGULAR'),
           (CODE#,AREA#,ACTIVITY#,VALUE_1,VALUE_2) FIXED BINARY;
         AREA# = VALUE_1;
         ACTIVITY# = VALUE_2;
         IF CODE# > 2 THEN DO;
           INITIAL = 1;
           LIMIT = 98;
           AREA#,ACTIVITY# = 1;
           END;
           ELSE IF CODE# = 2 THEN DO;
             INITIAL = ENTRY(AREA#) + 1;
             LIMIT = ENTRY(AREA# + 1);
             ACTIVITY# = 1;
             END;
             ELSE INITIAL,LIMIT = ENTRY(AREA#) + ACTIVITY#;
         PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(4))(SKIP(SPACE),
           COL(32),A);
         DO NUM = INITIAL TO LIMIT;
           /*

           IF FACULTY IS EQUAL TO ZERO, WHICH OCCURS ONLY WHEN
           THE ACTIVITY HAS NOT BEEN COMPLETED, DO NOT PRINT THIS
           ACTIVITY.
                                                                  */
           IF LEARNING_A(NUM).FACULTY = 0 THEN GO TO CHECK;
           /*

           SET THE CORRECT GROUP AND SUBSTITUTE STATUS VALUES.
                                                                  */
           IF LEARNING_A(NUM).GROUP = '1'B THEN G_STATUS = 1;
             ELSE G_STATUS = 2;
           IF LEARNING_A(NUM).SUBSTITUTE = '1'B THEN S_STATUS = 1
             ;
```

```
            ELSE S_STATUS = 2;
          PUT FILE(SYSPRINT) EDIT('AREA = ',AREA_DICT(AREA#),
            'NO = ',ACTIVITY#,SUBSTR(LEARNING_A(NUM).DATE,1,2)
            || '/' || SUBSTR(LEARNING_A(NUM).DATE,3,2) || '/' ||
            SUBSTR(LEARNING_A(NUM).DATE,5,2),FACULTY(LEARNING_A
            (NUM).FACULTY),ACTIVITY_TITLE_DICT(ACTIVITY_TITLE
            (NUM)),GP(G_STATUS),SUB(S_STATUS))(COL(46),
            2 A,X(3),A,F(2),X(3),A,X(2),A,COL(51),2 (A,X(5)),
            COL(51),A);
          PUT FILE(SYSPRINT) EDIT('COMMENTS =')(COL(51),A);
          DO N = 1 TO 6 WHILE(COMMENT(NUM,N) ¬= 0);
            PUT FILE(SYSPRINT) EDIT(COMMENT_DICT(COMMENT(NUM,N))
              )(COL(66),A);
            END;
          /*

          INCREMENT ACTIVITY# BY ONE.
                                                                */
CHECK:    ACTIVITY# = ACTIVITY# + 1;
          /*

          IF THE VALUE OF THE NEW INDEX INTO LEARNING_A IS
          EQUAL TO THE VALUE OF THE NEXT ELEMENT IN ENTRY,
          INCREMENT THE AREA# BY ONE AND RESET ACTIVITY# TO ONE.
                                                                */
          IF (ACTIVITY# + ENTRY(AREA#)) = ENTRY(AREA# + 1) + 1
            THEN DO;
            AREA# = AREA# + 1;
            ACTIVITY# = 1;
            END;
          END;
        SPACE = 3;
        END SUB_4;
        /*

        PRINT SUMMARY INFORMATION, WHICH CONSISTS OF THE TOTAL
        NUMBER OF LEARNING ACTIVITIES, THE PERCENT OF ACTIVITIES
        COMPLETED AND THE NUMBER OF THE MISSING LEARNING
        ACTIVITIES BY AREA AND OVERALL, FOR THE STUDENT'S
        LEARNING ACTIVITIES.
                                                                */
SUB_5:  PROCEDURE(CODE#,VALUE);
        DECLARE
          (CODE#,VALUE) FIXED BINARY;
        /*

        IF CODE# IS EQUAL TO TWO, PRINT SUMMARY INFORMATION
        ABOUT LEARNING ACTIVITIES IN ALL AREAS.  OTHERWISE,
        PRINT THIS INFORMATION FOR ONE AREA USING VALUE AS AN
        INDEX.
                                                                */
        IF CODE# = 2 THEN DO;
          INITIAL = 1;
          LIMIT = 10;
          END;
          ELSE INITIAL,LIMIT = VALUE;
        PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(5))(SKIP(SPACE),
          COL(32),A);
        TOTAL_ACTIVITIES,TOTAL_MISSING = 0;
        /*

        CALCULATE AND PRINT THE SUMMARY TOTALS BY AREA.  ALSO
        CALCULATE THE OVERALL SUMMARY TOTALS.
                                                                */
        DO NUM = INITIAL TO LIMIT;
          ACTIVITIES,MISSING = 0;
          DO KOUNT = ENTRY(NUM) TO (ENTRY(NUM + 1) - 2);
            ACTIVITIES = ACTIVITIES + 1;
            IF LEARNING_A(KOUNT + 1).DATE = ' ' THEN MISSING =
              MISSING + 1;
            END;
          TOTAL_ACTIVITIES = TOTAL_ACTIVITIES + ACTIVITIES;
          TOTAL_MISSING = TOTAL_MISSING + MISSING;
          PUT FILE(SYSPRINT) EDIT('AREA = ',AREA_DICT(NUM),
            'COMPLETED ',((ACTIVITIES - MISSING)/ACTIVITIES) *
            100,'% OF ',ACTIVITIES,' ACTIVITIES','MISSING ',
            MISSING,' ACTIVITIES')(COL(46),2 A,X(6),A,F(3),A,
            F(2),A,COL(90),A,F(2),A);
          END;
        /*

        IF CODE# IS EQUAL TO TWO, INCLUDE BEYOND THE MINIMUM
        ACTIVITIES IN THE OVERALL SUMMARY TOTALS AND PRINT
        THESE TOTALS.
```

```
                                                                      */
         IF CODE# = 2 THEN DO;
           DO KOUNT = ENTRY(11) TO (ENTRY(12) - 2);
             TOTAL_ACTIVITIES = TOTAL_ACTIVITIES + 1;
             IF LEARNING_A(KOUNT + 1).DATE = ' ' THEN
               TOTAL_MISSING = TOTAL_MISSING + 1;
             END;
           PUT FILE(SYSPRINT) EDIT('ALL LEARNING ACTIVITIES',
             'COMPLETED ',((TOTAL_ACTIVITIES - TOTAL_MISSING)/
             TOTAL_ACTIVITIES) * 100,'% OF ',TOTAL_ACTIVITIES,
             ' ACTIVITIES','MISSING ',TOTAL_MISSING,' ACTIVITIES'
             )(COL(46),A,COL(84),A,F(3),A,F(2),A,COL(90),A,
             F(2),A);
           END;
         SPACE = 3;
         END SUB_5;
         /*

         PRINT ALL INFORMATION CONTAINED IN THE STRUCTURE
         PERSONAL ABOUT THE STUDENT'S FIELD EXPERIENCE OR
         EXPERIENCES.
                                                                      */
SUB_6:   PROCEDURE(CODE#,VALUE);
         DECLARE
           (CODE#,VALUE) FIXED BINARY;
         /*

         IF CODE# IS EQUAL TO TWO, PRINT INFORMATION ABOUT ALL
         COMPLETED FIELD EXPERIENCES.  OTHERWISE, PRINT
         INFORMATION ABOUT ONE FIELD EXPERIENCE USING VALUE AS AN
         INDEX.
                                                                      */
         IF CODE# = 2 THEN DO;
           INITIAL = 1;
           LIMIT = 7;
           END;
           ELSE INITIAL,LIMIT = VALUE;
         PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(6))(SKIP(SPACE),
           COL(32),A);
         DO NUM = INITIAL TO LIMIT WHILE(FIELD.SCHOOL(NUM) ¬= 0);
           PUT FILE(SYSPRINT) EDIT('YR = ',FIELD(NUM).YR,'QTR = '
             ,FIELD(NUM).QTR,SCHOOL_DICT(FIELD(NUM).SCHOOL),
             FIELD(NUM).TEACHER,'GRADE = ',GRADE_DICT(FIELD(NUM).
             GRADE),CLASSROOM_DICT(FIELD(NUM).CLASSROOM),
             LEVEL_DICT(FIELD(NUM).LEVEL))(COL(46),A,F(2),X(3),
             A,F(1),X(3),A,X(1),A,COL(51),A,A,X(5),A,X(3),A);
           PUT FILE(SYSPRINT) EDIT('ROLE =')(COL(51),A);
           DO N = 1 TO 6 WHILE(F_ROLE(NUM,N) ¬= 0);
             PUT FILE(SYSPRINT) EDIT(F_ROLE_DICT(F_ROLE(NUM,N)))
               (COL(66),A);
             END;
           PUT FILE(SYSPRINT) EDIT('STRENGTHS = ')(COL(51),A);
           DO N = 1 TO 6 WHILE(STRENGTHS (NUM,N) ¬= 0);
             PUT FILE(SYSPRINT) EDIT(STRENGTHS_DICT(STRENGTHS
               (NUM,N)))(COL(66),A);
             END;
           PUT FILE(SYSPRINT) EDIT('NEEDS =')(COL(51),A);
           DO N = 1 TO 6 WHILE(WEAKNESSES(NUM,N) ¬= 0);
             PUT FILE(SYSPRINT) EDIT(WEAKNESSES_DICT(WEAKNESSES
               (NUM,N)))(COL(66),A);
             END;
           END;
         SPACE = 3;
         END SUB_6;
         /*

         PRINT SUMMARY INFORMATION, WHICH CONSISTS OF THE SCHOOL
         NAME AND GRADE LEVEL, ABOUT THE STUDENT'S FIELD
         EXPERIENCES.
                                                                      */
SUB_7:   PROCEDURE(CODE#,VALUE);
         DECLARE
           (CODE#,VALUE) FIXED BINARY;
         /*

         IF CODE# IS EQUAL TO TWO, PRINT A SUMMARY OF ALL FIELD
         EXPERIENCES COMPLETED.  OTHERWISE, PRINT A SUMMARY OF
         ONE FIELD EXPERIENCE USING VALUE AS AN INDEX.
                                                                      */
         IF CODE# = 2 THEN DO;
           INITIAL = 1;
           LIMIT = 7;
           END;
           ELSE INITIAL,LIMIT = VALUE;
         PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(7))(SKIP(SPACE),
           COL(32),A);
         DO NUM = INITIAL TO LIMIT WHILE(FIELD.SCHOOL(NUM) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(SCHOOL_DICT(FIELD(NUM).SCHOOL)
             ,'GRADE = ',GRADE_DICT(FIELD(NUM).GRADE),
             CLASSROOM_DICT(FIELD(NUM).CLASSROOM))(COL(46),A,X(1)
             ,2 A,X(5),A);
           END;
         SPACE = 3;
         END SUB_7;
         /*

         PRINT ALL INFORMATION CONTAINED IN THE STRUCTURE
         PERSONAL ABOUT THE STUDENT'S COMMUNITY EXPERIENCE.
                                                              */
SUB_8:   PROCEDURE;
         PUT FILE(SYSPRINT) EDIT(REQUEST_DICT(8))(SKIP(SPACE),
           COL(32),A);
         IF COMMUNITY.AGENCY = ' ' THEN RETURN;
         PUT FILE(SYSPRINT) EDIT('YR = ',COMMUNITY.YR,'QTR = ',
           COMMUNITY.QTR,COMMUNITY.AGENCY,'AGENCY REP = ',
           AGENCY_REP,LEVEL_DICT(COMMUNITY.LEVEL))(COL(46),
           A,F(2),X(3),A,F(1),X(3),A,X(5),A,X(3),A,COL(46),A);
         PUT FILE(SYSPRINT) EDIT('ROLE =')(COL(51),A);
         DO N = 1 TO 4 WHILE(C_ROLE(N) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(C_ROLE_DICT(C_ROLE(N)))
             (COL(66),A);
           END;
         PUT FILE(SYSPRINT) EDIT('COMMENTS =')(COL(51),A);
         DO N = 1 TO 4 WHILE(C_COMMENT(N) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(C_COMMENT_DICT(C_COMMENT(N)))
           (COL(66),A);
           END;
         SPACE = 3;
         END SUB_8;
FINISH:  END;
```

```
N_REG:  PROCEDURE OPTIONS(MAIN);
        /*
        PROGRAM:        PURGE

        OUTPUT
        DESCRIPTION:    ASCENDING LIST BY SOCIAL SECURITY NUMBER
                        OF STUDENTS WHOSE RECORDS WERE PURGED
                        FROM THE MAIN FILE AND SAVED UNLESS
                        OTHERWISE INDICATED.

        ALGORITHM:      READ THE INITIAL DATA.  PROCESS THE MAIN
                        FILE, BY STORING PURGED RECORDS IN THE
                        SAVE FILE AND CREATING A NEW MAIN FILE.
                        PRINT AN APPROPRIATE MESSAGE FOR EACH
                        RECORD TO BE PURGED.

        ******************************************************

        NOTE            THE FOLLOWING DECLARE STATEMENT(S)
                        INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                        ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                             1.  THE STATEMENT LABELED ONE WHICH
                                 CONTAINS THE STRUCTURE PERSONAL.

        ******************************************************
                                                              */
        DECLARE
          /*
          BIT WHEN SET TO A '1'B INDICATES THAT THE END OF A
          FILE HAS BEEN ENCOUNTERED ON AN INPUT DEVICE.        */
          BIT BIT(1),
          CUR_DATE CHARACTER(8),
          (OLD,NEW,SAVE) FILE RECORD SEQUENTIAL,
          SS_NO CHARACTER(9);
        /*

        INITIALIZE BIT TO A '0'B.
                                                              */
        BIT = '0'B;
        /*

        READ THE CURRENT DATE.
                                                              */
        GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
        /*

        ON ENCOUNTERING AN END OF FILE ON SYSIN, WHICH
        INDICATES THAT THERE ARE NO MORE RECORDS TO BE
        PURGED FROM THE MAIN FILE, SET BIT TO A '1'B.  COPY
        THE REMAINING RECORDS ONTO THE NEW MAIN FILE.
                                                              */
        ON ENDFILE(SYSIN) BEGIN;
          IF BIT = '1'B THEN GO TO FINISH;
          BIT = '1'B;
OTHERS:   READ FILE(OLD) INTO(PERSONAL);
          WRITE FILE(NEW) FROM(PERSONAL);
          GO TO OTHERS;
          END;
        /*

        ON ENCOUNTERING AN END OF FILE ON OLD, WHICH INDICATES
        THAT NO MORE RECORDS EXIST IN THE MAIN FILE, SET BIT TO
        A '1'B.  PRINT THE REMAINING SOCIAL SECURITY NUMBERS
        TO BE PURGED WITH THE MESSAGE 'MISSING.'
                                                              */
        ON ENDFILE(OLD) BEGIN;
          IF BIT = '1'B THEN GO TO FINISH;
          BIT = '1'B;
MESSAGE:  PUT FILE(SYSPRINT) EDIT('MISSING',SS_NO)(SKIP,A,X(12),
            A);
          GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
          GO TO MESSAGE;
          END;
        /*

        ON ENCOUNTERING THE END OF A PAGE, PRINT THE HEADER
        INFORMATION.
                                                              */
        ON ENDPAGE(SYSPRINT) BEGIN;
          PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
          PUT FILE(SYSPRINT) EDIT('PURGE',
            'OUTPUT DESCRIPTION: ASCENDING LIST OF STUDENTS ',
             'WHOSE RECORDS WERE PURGED',' FROM THE MAIN FILE ',
             'AND SAVED UNLESS OTHERWISE INDICATED')(SKIP,A,SKIP,
             2 A,COL(5),2 A);
           PUT FILE(SYSPRINT) EDIT('ERROR MESSAGE','S.S.NUMBER',
             'NAME','______________','___________','_____')(SKIP(6),
             A,COL(20),A,COL(40),A,SKIP(0),A,COL(20),A,COL(40),A)
             ;
           END;
         /*

         PRINT THE HEADER INFORMATION FOR THE FIRST PAGE OF
         PRINT.
                                                                  */
         PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
         PUT FILE(SYSPRINT) EDIT('PURGE',
           'OUTPUT DESCRIPTION: ASCENDING LIST OF STUDENTS ',
           'WHOSE RECORDS WERE PURGED',' FROM THE MAIN FILE ',
           'AND SAVED UNLESS OTHERWISE INDICATED')(SKIP,A,SKIP,
           2 A,COL(5),2 A);
         PUT FILE(SYSPRINT) EDIT('ERROR MESSAGE','S.S.NUMBER',
           'NAME','______________','___________','_____')(SKIP(6),
           A,COL(20),A,COL(40),A,SKIP(0),A,COL(20),A,COL(40),A);
         /*

         READ THE SOCIAL SECURITY NUMBER OF A RECORD TO BE
         PURGED FROM THE MAIN FILE.
                                                                  */
NEXT:    GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
         /*

         READ A RECORD FORM THE MAIN FILE.
                                                                  */
LOOP:    READ FILE(OLD) INTO(PERSONAL);
         /*

         IF THE RECORD TO BE PURGED HAS NOT YET BEEN FOUND IN
         THE MAIN FILE, COPY THE CURRENT RECORD ONTO THE NEW
         MAIN FILE.
                                                                  */
CHECK:   IF PERSONAL.SOC_SEC_NO < SS_NO THEN DO;
           WRITE FILE(NEW) FROM(PERSONAL);
           GO TO LOOP;
           END;
         /*

         IF THE RECORD TO BE PURGED DOES NOT EXIST IN THE MAIN
         FILE, PRINT THE SOCIAL SECURITY NUMBER WITH THE
         MESSAGE 'MISSING.'  READ THE SOCIAL SECURITY NUMBER OF
         THE NEXT RECORD TO BE PURGED.
                                                                  */
         IF PERSONAL.SOC_SEC_NO > SS_NO THEN DO;
           PUT FILE(SYSPRINT) EDIT('MISSING',SS_NO)
             (SKIP,A,X(12),A);
           GET FILE(SYSIN) EDIT(SS_NO)(SKIP,A(9));
           GO TO CHECK;
           END;
         /*

         THE RECORD TO BE PURGED HAS BEEN FOUND.  PRINT THE
         SOCIAL SECURITY NUMBER AND NAME FROM THIS RECORD.  COPY
         THIS RECORD ONTO THE SAVE FILE)
                                                                  */
         PUT FILE(SYSPRINT) EDIT(PERSONAL.SOC_SEC_NO,
           SUBSTR(PERSONAL.LAST,1,INDEX(PERSONAL.LAST,' ')-1) ||
           ',' || PERSONAL.FIRST)(COL(20),A,X(11),A);
         WRITE FILE(SAVE) FROM(PERSONAL);
         GO TO NEXT;
FINISH:  END;
```

```
REQUEST:PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:        REQUEST

        OUTPUT
        DESCRIPTION:    LISTING OF PERSONAL INFORMATION ON
                        STUDENTS REQUESTING FIELD EXPERIENCE.

        ALGORITHM:      READ THE INITIAL DATA.  PROCESS THE FILE
                        SEQUENTIALLY.  PRINT FIELD REQUEST
                        INFORMATION FOR STUDENTS IN THE SEMINARS
                        BEING PROCESSED.

        ************************************************************

        NOTE            THE FOLLOWING DECLARE STATEMENT(S)
                        INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                        ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                             1.   THE STATEMENT LABELED ONE WHICH
                                  CONTAINS THE STRUCTURE PERSONAL.
                             2.   THE STATEMENT LABELED TWO WHICH
                                  CONTAINS THE DICTIONARIES.

        ************************************************************

                                                                  */
        /*

        READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
        USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
        FACULTY_NO.
                                                                  */
        GET FILE(SYSIN) EDIT(MIN,MAX)(COL(1),2 (F(1),X(4)));
        BEGIN;
        DECLARE
          /*
          DICTIONARY OF THE FACULTY MEMBER'S NAMES ASSOCIATED
          WITH THE CODES USED IN PERSONAL.                        */
          FACULTY(25) CHARACTER(14) INITIAL('BERNARD','CURRAN',
             'DETURE','FEY','GOLDMAN','HILLIARD','JESTER',
             'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
             'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
             'TISON','WILMOT','KRALL','CREEDON','LEVINE',
             'HEDGES','GRIMM'),
          /*
          FACULTY NUMBERS OF THE SEMINAR LEADERS BY TEAM.         */
          FACULTY_NO(MIN:MAX,4) FIXED BINARY,
          /*
          DATE OF THE UPCOMING FIELD EXPERIENCE.                  */
          REQUEST_DATE CHARACTER(15),
          CUR_DATE CHARACTER(8),
          (LOCAL_LENGTH,PERM_LENGTH) FIXED BINARY,
          OLD FILE RECORD SEQUENTIAL;
        /*

        INITIALIZE FACULTY_NO.
                                                                  */
        FACULTY_NO = 0;
        /*

        ON ENCOUNTERING THE END OF THE MAIN FILE TRANSFER TO
        THE LABEL FINISH.
                                                                  */
        ON ENDFILE(OLD) GO TO FINISH;
        OPEN FILE(SYSPRINT) LINESIZE(130);
        /*

        READ THE CURRENT DATE.  THEN READ THE DATE, WHICH
        CONSISTS OF THE YEAR AND QUARTER, OF THE UPCOMING FIELD
        EXPERIENCE.
                                                                  */
        GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
        GET FILE(SYSIN) EDIT(REQUEST_DATE)(COL(1),A(15));
        /*

        READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
        TEAM READ THE TEAM NUMBER, THE NUMBER OF SEMINARS TO BE
        PROCESSED, AND THE SEMINAR NUMBER AND FACULTY NUMBER
        OF EACH SEMINAR TO BE PROCESSED.
                                                                  */
        GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
        DO I = 1 TO NO_CARDS;
          GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
            TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
            X(4)),4 (F(I),X(4),F(2),X(3)));
            END;
         /*

         READ A STUDENT'S RECORD.
                                                                */
LOOP:    READ FILE(OLD) INTO(PERSONAL);
         /*

         IF THE STUDENT'S RECORD IS NOT TO BE PROCESSED, OBTAIN
         A NEW RECORD
                                                                */
         IF (FACULTY_NO(TEAM,SEMINAR) = 0) THEN GO TO LOOP;
         /*

         PRINT THE HEADER INFORMATION.
                                                                */
         PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
         PUT FILE(SYSPRINT) EDIT('REQUEST','OUTPUT DESCRIPTION:',
           ' LISTING OF PERSONAL INFORMATION ON STUDENTS ',
           'REQUESTING FIELD EXPERIENCE')(SKIP,A,SKIP,3 A);
         PUT FILE(SYSPRINT) EDIT('REQUEST FOR ',REQUEST_DATE)(SKIP,2 A);
         /*

         PRINT THE STUDENT'S NANE.
                                                                */
         PUT FILE(SYSPRINT) EDIT('NAME:',SUBSTR(PERSONAL.FIRST,1,
           INDEX(PERSONAL.FIRST,' ') - 1) || ' ' ||
           PERSONAL.LAST)(SKIP(6),A,COL(25),A);
         /*

         CALCULATE THE LENGTH OF CITY FOR THE PERMANENT ADDRESS
         AND THE LOCAL ADDSESS.
                                                                */
         PERM_LENGTH = INDEX(PERM_ADDR.CITY,'  ');
         IF PERM_LENGTH = 0 THEN PERM_LENGTH = 20;
         LOCAL_LENGTH = INDEX(LOCAL_ADDR.CITY,'  ');
         IF LOCAL_LENGTH = 0 THEN LOCAL_LENGTH = 20;
         /*

         PRINT THE STUDENT'S LOCAL ADDRESS AND PERMANENT ADDRESS.
                                                                */
         PUT FILE(SYSPRINT) EDIT('LOCAL ADDRESS:',
           LOCAL_ADDR.STREET,'HOME ADDRESS:',PERM_ADDR.STREET,
           SUBSTR(LOCAL_ADDR.CITY,1,LOCAL_LENGTH
           -1),', ',LOCAL_ADDR.STATE,LOCAL_ADDR.ZIP,
           SUBSTR(PERM_ADDR.CITY,1,PERM_LENGTH
           -1),', ',PERM_ADDR.STATE,PERM_ADDR.ZIP,
           LOCAL_PHONE,PERM_PHONE)(SKIP(2),A,COL(25),A,COL(65),A,
           COL(90),A,COL(25),3 A,X(2),A,COL(90),3 A,X(2),A,
           COL(25),A,COL(90),A);
         /*

         PRINT THE STUDENT'S DATE OF BIRTH.
                                                                */
         PUT FILE(SYSPRINT) EDIT('DATE OF BIRTH:',
           BIRTH_DATE.MONTH,'/',BIRTH_DATE.DAY,'/',BIRTH_DATE.YR)
           (SKIP(2),A,COL(25),F(2),A,F(2),A,F(2));
         /*

         PRINT THE STUDENT'S TRAVEL AND WORK EXPERIENCES.
                                                                */
         PUT FILE(SYSPRINT) EDIT('TRAVEL EXPERIENCE:',
           'WORK EXPERIENCE:')(SKIP(2),A,COL(65),A);
         PUT SKIP(0);
         DO N = 1 TO 6 WHILE(TRAVEL_EXP(N) ¬= 0 |
           WORK_EXP(N) ¬= 0);
           PUT FILE(SYSPRINT) EDIT(TRAVEL_EXP_DICT(TRAVEL_EXP(N))
              ,WORK_EXP_DICT(WORK_EXP(N)))(CO_(25),A,COL(90),A);
           END;
         /*

         PRINT INFORMATION ABOUT THE STUDENT'S CURRENT
         EMPLOYMENT.
                                                                */
         PUT FILE(SYSPRINT) EDIT('CURRENT EMPLOYMENT:',EMPLOYER,
           'HOURS PER WEEK =',HRS_PER_WK)(SKIP(2),A,COL(25),A,
           COL(65),A,X(3),F(2));
         /*

         PRINT THE NAME OF THE STUDENT'S SEMINAR LEADER.
                                                                */
         PUT FILE(SYSPRINT) EDIT('SEMINAR LEADER:',FACULTY(
          FACULTY_NO(TEAM,SEMINAR)))(SKIP(2),A,COL(25),A);
        /*

        PRINT THE STUDENT'S AREA OF SPECIALIZATION
                                                                   */
        PUT FILE(SYSPRINT) EDIT('SPECIALIZATION:',
          SPECIALIZATION_DICT(SPECIALIZATION))(SKIP(2),A,COL(25)
          ,A);
        /*

        PRINT INFORMATION ABOUT THE STUDENT'S PREVIOUS FIELD
        EXPERIENCES.
                                                                   */
        PUT FILE(SYSPRINT) EDIT('FIELD EXPERIENCES',
          'COMPLETED TO DATE:')(SKIP(2),A,SKIP,A);
        DO NUM = 1 TO 7 WHILE(FIELD.SCHOOL(NUM) ¬= 0);
          PUT FILE(SYSPRINT) EDIT('YR =',FIELD(NUM).YR,'QTR =',
            FIELD(NUM).QTR,'LEVEL =',LEVEL_DICT(FIELD(NUM).LEVEL
            ),'SCHOOL =',SCHOOL_DICT(FIELD(NUM).SCHOOL),
            'TEACHER =',FIELD(NUM).TEACHER,'GRADE =',GRADE_DICT
            (FIELD(NUM).GRADE))(COL(25),A,F(2),X(3),A,F(1),X(3),
            A,X(3),A,COL(30),3 (A,X(3),A,X(3)));
          PUT FILE(SYSPRINT) EDIT('ROLE =')(COL(30),A);
          DO N = 1 TO 6 WHILE(F_ROLE(NUM,N) ¬= 0);
            PUT FILE(SYSPRINT) EDIT(F_ROLE_DICT(F_ROLE(NUM,N)))
              (COL(45),A);
            END;
          PUT FILE(SYSPRINT) EDIT('STRENGTHS =')(COL(30),A);
          DO N = 1 TO 6 WHILE(STRENGTHS(NUM,N) ¬= 0);
            PUT FILE(SYSPRINT) EDIT(STRENGTHS_DICT(STRENGTHS
              (NUM,N)))(COL(45),A);
            END;
          PUT FILE(SYSPRINT) EDIT('NEEDS =')(COL(30),A);
          DO N = 1 TO 6 WHILE(WEAKNESSES(NUM,N) ¬= 0);
          PUT FILE(SYSPRINT) EDIT(WEAKNESSES_DICT(WEAKNESSES
              (NUM,N)))(COL(45),A);
            END;
          END;
        GO TO LOOP;
        END;
FINISH: END;
```

```
S_LIST: PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:       SS

        OUTPUT
        DESCRIPTION:   ALPHABETICAL LIST OF TEAM MEMBERS AND
                       THEIR SOCIAL SECURITY NUMBERS BY SEMINAR.

        ALGORITHM:     READ THE INITIAL DATA.  PROCESS THE MAIN
                       FILE BY STORING THE NAME AND SOCIAL
                       SECURITY NUMBER OF STUDENTS IN THE
                       SEMINARS BEING PROCESSED IN THE STRUCTURE
                       STORE.  ON ENCOUNTERING THE END OF THE
                       MAIN FILE, SORT THE SEMINARS IN THE
                       STRUCTURE STORE ALPHABETICALLY BY NAME.
                       ELIMINATE EXCESS BLANKS FROM THE NAMES IN
                       THE STRUCTURE STORE.  PRINT THE REQUESTED
                       TEAM LISTS BY SEMINAR.

        ********************************************************

        NOTE           THE FOLLOWING DECLARE STATEMENT(S)
                       INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                       ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                            1.  THE STATEMENT LABELED ONE WHICH
                                CONTAINS THE STRUCTURE PERSONAL.

        ********************************************************

                                                             */
        /*

        READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
        USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
        BIT, FACULTY_NO, STORE AND VALUE.
                                                             */
        GET FILE(SYSIN) EDIT(MIN,MAX)(COL(1),2 (F(1),X(4)));
        BEGIN;
        DECLARE
          /*
          INDICATOR OF THE PRINT STATUS FOR EACH TEAM.       */
          BIT(MIN:MAX) BIT(1),
          /*
          DICTIONARY OF THE FACULTY MEMBER'S NAMES ASSOCIATED
          WITH THE CODES USED IN PERSONAL.                   */
          FACULTY(20) CHARACTER(14) INITIAL('BERNARD','CURRAN',
            'DETURE','FEY','GOODMAN','HILLIARD','JESTER',
            'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
            'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
            'TISON','WILMOT'),
          /*
          SEMINAR LEADER'S FACULTY NUMBERS BY TEAM.          */
          FACULTY_NO(MIN:MAX,4) FIXED BINARY,
          /*
          NAME AND SOCIAL SECURITY NUMBER OF STUDENTS IN EACH
          TEAM BY SEMINAR.                                   */
          1 STORE(MIN:MAX,4,35),
            2 NAMES CHARACTER(25),
            2 SS_NO CHARACTER(9),
          /*
          COUNTER FOR THE NUMBER OF STUDENTS IN EACH SEMINAR BY
          TEAM.                                              */
          VALUE(MIN:MAX,4) FIXED BINARY,
          CUR_DATE CHARACTER(8),
          N_TEMP CHARACTER(25) VARYING,
          OLD FILE RECORD SEQUENTIAL,
          SEMINAR# FIXED BINARY,
          TEAM# FIXED BINARY,
          1 TEMP,
            2 T_NAMES CHARACTER(25),
            2 T_SS_NO CHARACTER(9);
        /*

        INITIALIZE STORE, VALUE AND FACULTY_NO.

                                                             */
        STORE = '0';
        VALUE = 0;
        FACULTY_NO = 0;
        /*

        ON ENCOUNTERING AN ILLEGAL SUBSCRIPT IN A STUDENT'S
        RECORD, PRINT THE STUDENT'S SOCIAL SECURITY NUMBER, TEAM
```

```
         NUMBER, SEMINAR NUMBER AND THE TOTAL NUMBER OF STUDENTS
         IN THIS SEMINAR.
                                                                 */
         ON SUBSCRIPTRANGE BEGIN;
           PUT FILE(SYSPRINT) EDIT('SUBSCRIPT OUTSIDE LEGAL',
             ' BOUNDS','SOC_SEC_NO = ',SOC_SEC_NO,'TEAM = ',TEAM,
             'SEMINAR = ',SEMINAR,'VALUE = ',VALUE(TEAM,SEMINAR))
             (2 A,SKIP(3),A,A,X(5),3 (A,F(2),X(5)));
           GO TO FINISH;
           END;
         /*

         ON ENCOUNTERING THE END OF THE MAIN FILE TRANSFER TO THE
         LABEL SORT.
                                                                 */
         ON ENDFILE(OLD) GO TO SORT;
         OPEN FILE(SYSPRINT) LINESIZE(130);
         /*

         READ THE CURRENT DATE.
                                                                 */
         GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
         /*

         READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
         TEAM READ THE TEAM NUMBER. THE NUMBER OF SEMINARS TO BE
         PROCESSED, AND THE SEMINAR NUMBER AND FACULTY NUMBER
         OF EACH SEMINAR TO BE PROCESSED.
                                                                 */
         GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
         DO I = 1 TO NO_CARDS;
           GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
             TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
             X(4)),4 (F(1),X(4),F(2),X(3)));
             END;
         /*

         PROCESS THE FILE SEQUENTIALLY.  READ A STUDENT'S RECORD.
         IF THIS RECORD IS NOT TO BE PRINTED, TRANSFER TO LOOP TO
         GET A NEW RECORD.  INCREMENT THE CORRECT COUNTER IN
         VALUE BY ONE.  STORE THE STUDENT'S NAME AND SOCIAL
         SECURITY NUMBER IN THE CORRECT POSITION IN THE ARRAY
         NAMES.
                                                                 */
LOOP:    READ FILE(OLD) INTO(PERSONAL);
         IF TEAM < MIN | TEAM > MAX THEN GO TO LOOP;
         IF FACULTY_NO(TEAM,SEMINAR) = 0 THEN GO TO LOOP;
         VALUE(TEAM,SEMINAR) = VALUE(TEAM,SEMINAR) + 1;
         NAMES(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) = LAST || ',' ||
           FIRST;
         SS_NO(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) = SOC_SEC_NO;
         GO TO LOOP;
         /*

         SET BIT TO A '1'B IF THIS TEAM IS NOT TO BE PRINTED.
         SORT THE SEMINARS IN THE ARRAY NAME ALPHABETICALLY BY
         NAME.
                                                                 */
SORT:    DO TEAM# = MIN TO MAX;
           BIT(TEAM#) = FACULTY_NO(TEAM#,1) = 0 &
             FACULTY_NO(TEAM#,2) = 0 & FACULTY_NO(TEAM#,3) = 0 &
             FACULTY_NO(TEAM#,4) = 0;
           IF BIT(TEAM#) = '1'B THEN GO TO T_END;
           DO SEMINAR# = 1 TO 4;
             IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO S_END;
             DO I = 1 TO 34;
               DO J = I+1 TO 35;
                 IF NAMES(TEAM#,SEMINAR#,I) >
                   NAMES(TEAM#,SEMINAR#,J) THEN DO;
                   TEMP = STORE(TEAM#,SEMINAR#,I);
                   STORE(TEAM#,SEMINAR#,I) = STORE(TEAM#,SEMINAR#
                     ,J);
                   STORE(TEAM#,SEMINAR#,J) = TEMP;
                   END;
               END;
             END;
S_END:     END;
T_END:   END;
         /*

         PRINT A TEAM LIST WITH SOCIAL SECURITY NUMBERS FOR EACH
         TEAM REQUESTING IT.  PRINT ONE TEAM PER PAGE.
                                                                 */
         DO TEAM# = MIN TO MAX;
```

```
           IF BIT(TEAM#) = '1'B THEN GO TO OUTER;
           /*

           PRINT THE HEADER INFORMATION FOR EACH PAGE OF PRINT.
                                                               */
           PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
           PUT FILE(SYSPRINT) EDIT('SS',
             'OUTPUT DESCRIPTION: ALPHABETICAL LIST OF TEAM ',
             'MEMBERS AND THEIR SOCIAL SECURITY NUMBERS BY ',
             'SEMINAR','TEAM',TEAM#)(SKIP,A,SKIP,3 A,SKIP,A,X(2),
             F(2));
           PUT SKIP(10);
           NCNBLANK = 0;
           DO SEMINAR# = I TO 4;
             IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO INNER;
             NONBLANK = NONBLANK + 1;
             /*

             PRINT THE SEMINAR LEADER'S NAME.
                                                              */
             PUT FILE(SYSPRINT) EDIT(FACULTY(FACULTY_NO(TEAM#,
               SEMINAR#)))(X(10),A);
INNER:       END;
           PUT FILE(SYSPRINT) EDIT(('_______________' DO I = 1
             TO NCNBLANK))(SKIP(0),(NONBLANK)(X(10),A));
           PUT SKIP(2);
           /*

           PRINT BY SEMINAR IN ALPHABETICAL ORDER BY LAST NAME,
           THE NAMES AND SOCIAL SECURITY NUMBERS OF THE SEMINAR
           MEMBERS.
                                                              */
           DO NUM = 1 TO 35;
               PUT SKIP;
               KOUNT = 10;
               DO SEMINAR# = I TO 4;
               IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO
                 INNER2;
               IF NAMES(TEAM#,SEMINAR#,NUM) = '0' THEN DO;
                 KOUNT = KOUNT + 24;
                 GC TO INNER2;
                 END;
               /*

               ELIMINATE EXCESS BLANKS BETWEEN THE STUDENT'S LAST
               NAME AND THE ',' IN NAMES.
                                                              */
               N_TEMP = SUBSTR(NAMES(TEAM#,SEMINAR#,NUM),1,
                 INDEX(NAMES(TEAM#,SEMINAR#,NUM),' ') - 1) ||
                 SUBSTR(NAMES(TEAM#,SEMINAR#,NUM),INDEX(NAMES(
                 TEAM#,SEMINAR#,NUM),','));
               PUT FILE(SYSPRINT) EDIT(STORE.SS_NO(TEAM#,SEMINAR#,NUM),
                 N_TEMP)(X(KOUNT),A,X(5),A);
               KOUNT = 24 - LENGTH(N_TEMP);
INNER2:        END;
             END;
OUTER:     ENC;
        END;
FINISH: END;
```

```
LIST:   PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:          TEAM

        OUTPUT
        DESCRIPTION:      ALPHABETICAL LIST OF THE TEAM MEMBERS BY
                          SEMINAR)

        ALGORITHM:        READ THE INITIAL DATA.  PROCESS THE MAIN
                          FILE BY STORING THE NAMES OF STUDENTS
                          THE SEMINARS BEING PROCESSED IN THE ARRAY
                          NAMES.  ON ENCOUNTERING THE END OF
                          THE MAIN FILE, SORT THE SEMINARS IN THE
                          THE ARRAY NAMES ALPHABETICALLY.
                          ELIMINATE EXCESS BLANKS FROM THE NAMES IN
                          THE ARRAY NAMES.  PRINT THE REQUESTED
                          TEAM LIST BY SEMINAR.

        ***********************************************************

        NOTE              THE FOLLOWING DECLARE STATEMENT(S)
                          INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                          ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                                1.  THE STATEMENT LABELED ONE WHICH
                                    CONTAINS THE STRUCTURE PERSONAL.

        ***********************************************************
                                                                  */
        /*

        READ THE RANGE OF THE TEAM NUMBERS.  THIS RANGE IS
        USED TO DYNAMICALLY CONTROL THE STORAGE ALLOCATED TO
        BIT, FACULTY_NO, STORE AND VALUE.
                                                                  */
        GET FILE(SYSIN) EDIT(MIN,MAX)(COL(1),2 (F(1),X(4)));
        BEGIN;
        DECLARE
          /*
          INDICATOR OF THE PRINT STATUS FOR EACH TEAM.            */
          BIT(MIN:MAX) BIT(1),
          /*
          DICTIONARY OF THE FACULTY MEMBER'S NAMES ASSOCIATED
          WITH THE CODES USED IN PERSONAL.                        */
          FACULTY(20) CHARACTER(14) INITIAL('BERNARD','CURRAN',
             'DETURE','FEY','GOODMAN','HILLIARD','JESTER',
             'KINZER','LEWIS','LITCHER','MARTINELLO','NEWCOMB',
             'OLSON','PENLAND','ROWE','SERAPHINE','SHEA','SMITH',
             'TISON','WILMOT'),
          /*
          FACULTY NUMBERS OF THE SEMINAR LEADERS BY TEAM.         */
          FACULTY_NO(MIN:MAX,4) FIXED BINARY,
          /*
          NAMES OF THE STUDENTS IN EACH SEMINAR BY TEAM.          */
          NAMES(MIN:MAX,4,35) CHARACTER(25),
          /*
          COUNTER FOR THE NUMBER OF STUDENTS IN EACH SEMINAR BY
          TEAM.                                                   */
          VALUE(MIN:MAX,4) FIXED BINARY,
          CUR_DATE CHARACTER(8),
          OLD FILE RECORD SEQUENTIAL,
          SEMINAR# FIXED BINARY,
          TEAM# FIXED BINARY,
          TEMP CHARACTER(25) VARYING;
        /*

        INITIALIZE STORE, VALUE AND FACULTY_NO.
                                                                  */
        NAMES = '0';
        VALUE = 0;
        FACULTY_NO = 0;
        /*

        ON ENCOUNTERING AN ILLEGAL SUBSCRIPT IN A STUDENT'S
        RECORD, PRINT THE STUDENT'S SOCIAL SECURITY NUMBER, TEAM
        NUMBER, SEMINAR NUMBER AND THE TOTAL NUMBER OF STUDENTS
        IN THIS SEMINAR.
                                                                  */
        ON SUBSCRIPTRANGE BEGIN;
          PUT FILE(SYSPRINT) EDIT('SUBSCRIPT OUTSIDE LEGAL',
             ' BOUNDS','SOC_SEC_NO = ',SOC_SEC_NO,'TEAM = ',TEAM,
             'SEMINAR = ',SEMINAR,'VALUE = ',VALUE(TEAM,SEMINAR))
             (2 A,SKIP(3),A,A,X(5),3 (A,F(2),X(5)));
```

```
           GO TO FINISH;
           END;
         /*

         ON ENCOUNTERING THE END OF THE MAIN FILE TRANSFER TO
         SORT.
                                                               */
         ON ENDFILE(OLD) GO TO SORT;
         OPEN FILE(SYSPRINT) LINESIZE(130);
         /*

         READ THE CURRENT DATE.
                                                               */
         GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
         /*

         READ THE NUMBER OF TEAMS TO BE PROCESSED.  THEN FOR EACH
         TEAM READ THE TEAM NUMBER, THE NUMBER OF SEMINARS TO BE
         PROCESSED, AND THE SEMINAR NUMBER AND FACULTY NUMBER
         OF EACH SEMINAR TO BE PROCESSED.
                                                               */
         GET FILE(SYSIN) EDIT(NO_CARDS)(COL(1),F(2));
         DO I = 1 TO NO_CARDS;
           GET FILE(SYSIN) EDIT(TEAM#,LIMIT,(SEMINAR#,FACULTY_NO(
              TEAM#,SEMINAR#) DO J = 1 TO LIMIT))(COL(1),2 (F(1),
              X(4)),4 (F(1),X(4),F(2),X(3)));
              END;
         /*

         PROCESS THE FILE SEQUENTIALLY.  READ A STUDENT'S RECORD.
         IF THE RECORD IS NOT TO BE PRINTED TRANSFER TO LOOP
         TO GET A NEW RECORD.  INCREMENT THE CORRECT COUNTER IN
         VALUE BY ONE.  STORE THE STUDENT'S NAME IN THE CORRECT
         POSITION IN THE STRUCTURE NAMES.
                                                               */
LOOP:    READ FILE(OLD) INTO(PERSONAL);
         IF TEAM < MIN | TEAM > MAX THEN GO TO LOOP;
         IF FACULTY_NO(TEAM,SEMINAR) = 0 THEN GO TO LOOP;
         VALUE(TEAM,SEMINAR) = VALUE(TEAM,SEMINAR) + 1;
         NAMES(TEAM,SEMINAR,VALUE(TEAM,SEMINAR)) = LAST || ',' ||
           FIRST;
         GO TO LOOP;
         /*

         SET BIT TO A '1'B IF THIS TEAM IS NOT TO BE PRINTED.
         SORT THE SEMINARS IN THE STRUCTURE NAME ALPHABETICALLY
         BY NAME.
                                                               */
SORT:    DO TEAM# = MIN TO MAX;
           BIT(TEAM#) = FACULTY_NO(TEAM#,1) = 0 &
             FACULTY_NO(TEAM#,2) = 0 & FACULTY_NO(TEAM#,3) = 0 &
             FACULTY_NO(TEAM#,4) = 0;
           IF BIT(TEAM#) = '1'B THEN GO TO T_END;
           DO SEMINAR# = 1 TO 4;
             IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO S_END;
             DO I = 1 TO 34;
               DO J = I+1 TO 35;
                 IF NAMES(TEAM#,SEMINAR#,I) >
                   NAMES(TEAM#,SEMINAR#,J) THEN DO;
                   TEMP = NAMES(TEAM#,SEMINAR#,I);
                   NAMES(TEAM#,SEMINAR#,I) = NAMES(TEAM#,SEMINAR#
                     ,J);
                   NAMES(TEAM#,SEMINAR#,J) = TEMP;
                   END;
               END;
             END;
S_END:     END;
T_END:   END;
         /*

         PRINT A TEAM LIST FOR EACH TEAM REQUESTING IT.  LIST
         ONE TEAM PER PAGE.
                                                               */
         DO TEAM# = MIN TO MAX;
           IF BIT(TEAM#) = '1'B THEN GO TO OUTER;
           /*

           PRINT THE HEADER INFORMATION FOR EACH PAGE OF PRINT.
                                                               */
           PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
           PUT FILE(SYSPRINT) EDIT('TEAM',
             'OUTPUT DESCRIPTION: ALPHABETICAL LIST OF TEAM ',
             'MEMBERS BY SEMINAR','TEAM',TEAM#)(SKIP,A,SKIP,2 A,
             SKIP,A,X(2),F(2));
```

```
          PUT SKIP(10);
          NONBLANK = 0;
          DO SEMINAR# = 1 TO 4;
            IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO INNER;
            NONBLANK = NONBLANK + 1;
            /*

            PRINT THE SEMINAR LEADER'S NAME.
                                                                  */
            PUT FILE(SYSPRINT) EDIT(FACULTY(FACULTY_NO(TEAM#,
              SEMINAR#)))(X(10),A);
INNER:      END;
          PUT FILE(SYSPRINT) EDIT(('_______________'  DO I = 1
            TO NONBLANK))(SKIP(0),(NONBLANK)(X(10),A));
          PUT SKIP(2);
          /*

          PRINT BY SEMINAR IN ALPHABETICAL ORDER BY LAST NAME
          THE NAMES OF THE SEMINAR MEMBERS.
                                                                  */
          DO NUM = 1 TO 35;
              PUT SKIP;
              KOUNT = 10;
              DO SEMINAR# = 1 TO 4;
              IF FACULTY_NO(TEAM#,SEMINAR#) = 0 THEN GO TO
                INNER2;
              IF NAMES(TEAM#,SEMINAR#,NUM) = '0' THEN DO;
                KOUNT = KOUNT + 24;
                GO TO INNER2;
                END;
              /*

              ELIMINATE EXCESS BLANKS BETWEEN THE STUDENT'S LAST
              NAME AND THE ',' IN NAMES.
                                                                  */
              TEMP = SUBSTR(NAMES(TEAM#,SEMINAR#,NUM),1,
                INDEX(NAMES(TEAM#,SEMINAR#,NUM),' ') - 1) ||
                SUBSTR(NAMES(TEAM#,SEMINAR#,NUM),INDEX(NAMES(
                TEAM#,SEMINAR#,NUM),','));
              PUT FILE(SYSPRINT) EDIT(TEMP)(X(KOUNT),A);
              KOUNT = 24 - LENGTH(TEMP);
INNER2:       END;
            END;
OUTER:    END;
        END;
FINISH: END;
```

```
UPDATE: PROCEDURE OPTIONS(MAIN);
        /*

        PROGRAM:        UPDATE

        OUTPUT
        DESCRIPTION:    A LISTING BY SOCIAL SECURITY NUMBER OF
                        THE OLD AND NEW CONTENTS OF ALL ITEMS
                        WHICH HAVE BEEN UPDATED.

        ALGORITHM:      READ THE INITIAL DATA.  PROCESS THE MAIN
                        FILE SEQUENTIALLY.  UPDATE THE SPECIFIED
                        RECORDS AND BUILD A NEW UPDATED MAIN
                        FILE.

        **************************************************************

        NOTE            THE FOLLOWING DECLARE STATEMENT(S)
                        INCLUDED IN THE PROGRAM NAMED ALL MUST BE
                        ADDED TO THIS PROGRAM PRIOR TO EXECUTION:
                             I.  THE STATEMENT LABELED ONE WHICH
                                 CONTAINS THE STRUCTURE PERSONAL.

        **************************************************************
                                                                    */
        DECLARE
          /*
          BIT WHEN SET TO A '1'B INDICATES THAT AN END OF FILE
          HAS BEEN ENCOUNTERED ON AN INPUT DEVICE.                  */
          BIT BIT(1) INITIAL('0'B),
          /*
          AN ENTRY POINT, FOR INSTANCE ENTRY(1), INDICATES THE
          NUMBER OF THE POSITION PRECEEDING THE BEGINNING OF
          THIS SUBSTANTIVE AREA IN LEARNING_A.                      */
          ENTRY(11) FIXED BINARY INITIAL(0,6,19,31,41,50,55,66,
            82,87,95),
          /*
          AN ARRAY OF LABELS WHICH IS INITIALIZED TO THE UPDATE
          ROUTINES FOR THE VARIOUS ITEMS.                           */
          TRANSFER(86) LABEL INITIAL(NO_1,NO_2,NO_3,NO_4,NO_5,
            NO_6,NO_7,NO_8,NO_9,NO_10,NO_11,NO_12,NO_13,NO_14,
            NO_15,NO_16,NO_17,NO_18,NO_19,NO_20,NO_21,NO_22,
            NO_23,NO_24,NO_25,NO_26,NO_27,NO_28,NO_29,NO_30,
            NO_31,NO_32,NO_33,NO_34,NO_35,NO_36,NO_37,NO_38,
            NO_39,NO_40,NO_41,NO_42,NO_43,NO_44,NO_45,NO_46,
            NO_47,NO_48,NO_49,NO_50,NO_51,NO_52,NO_53,NO_54,NO_55,
            NO_56,NO_57,NO_58,NO_59,NO_60,NO_61,NO_62,NO_63,
            NO_64,NO_65,NO_66,NO_67,NO_68,NO_69,NO_70,NO_71,
            NO_72,NO_73,NO_74,NO_75,NO_76,NO_77,NO_78,NO_79,
            NO_80,NO_81,NO_82,NO_83,NO_84,NO_85,NO_86),
          AREA# FIXED BINARY,
          ACTIVITY# FIXED BINARY,
          CUR_DATE CHARACTER(8),
          ARRAY ENTRY (FIXED BINARY,(*) FIXED BINARY,
            FIXED BINARY),
          LETTER ENTRY (CHARACTER(*),FIXED BINARY),
          LOC LABEL,
          (NEW,OLD) FILE RECORD SEQUENTIAL,
          NUMBER ENTRY (FIXED BINARY,FIXED BINARY),
          POS FIXED BINARY,
          SS_NO CHARACTER(9),
          STORE_C CHARACTER(40) VARYING,
          STORE_A(18) FIXED BINARY,
          1 STORE_S(5),
            2 S_COURSE_NO CHARACTER(6),
            2 S_HRS FIXED BINARY,
            2 S_GRADE CHARACTER(1);
        OPEN FILE(SYSPRINT) LINESIZE(130);
        /*

        ON ENCOUNTERING AN END OF FILE ON OLD, WHICH INDICATES
        THAT THERE ARE NO MORE RECORDS TO BE UPDATED IN THE MAIN
        FILE, SET BIT TO A '1'B.  COPY THE REMAINING RECORDS
        ONTO THE NEW MAIN FILE.
                                                                    */
        ON ENDFILE(SYSIN) BEGIN;
          IF BIT = '1'B THEN GO TO FINISH;
          BIT = '1'B;
MORE:     READ FILE(OLD) INTO(PERSONAL);
          WRITE FILE(NEW) FROM(PERSONAL);
          GO TO MORE;
          END;
        /*
```

```
          ON ENCOUNTERING THE END OF A PAGE OF PRINT, PRINT THE
          HEADER INFORMATION.
                                                                   */
          ON ENDPAGE(SYSPRINT) BEGIN;
            PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
            PUT FILE(SYSPRINT) EDIT('UPDATE',
              'OUTPUT DESCRIPTION: A LISTING BY SOCIAL SECURITY',
              ' NUMBER OF THE OLD AND NEW CONTENTS OF ALL ITEMS',
              ' WHICH HAVE BEEN UPDATED')(SKIP,A,SKIP,3 A);
            PUT FILE(SYSPRINT) EDIT('SOCIAL SECURITY NO','ITEM',
              'PREVIOUS CONTENTS','PRESENT CONTENTS')
              (SKIP(6),A,COL(25),A,COL(35),A,COL(85),A);
            PUT FILE(SYSPRINT) EDIT('__________________','____',
              '_________________________','____________________')
              (SKIP(0),COL(1),A,COL(25),A,COL(35),A,COL(85),A);
            END;
          /*

          ON ENCOUNTERING A SUBSCRIPTRANGE ERROR, PRINT AN ERROR
          MESSAGE.  TRANSFER TO OBTAIN A NEW UPDATE NUMBER.
                                                                   */
          ON SUBSCRIPTRANGE BEGIN;
            PUT FILE(SYSPRINT) EDIT('SUBSCRIPT')(SKIP(0),A);
            GO TO NEXT;
            END;
          /*

          ON ENCOUNTERING A CONVERSION ERROR, PRINT AN ERROR
          MESSAGE.  TRANSFER TO OBTAIN A NEW UPDATE NUMBER.
                                                                   */
          ON CONVERSION BEGIN;
            PUT FILE(SYSPRINT) EDIT('CONVERSION')(SKIP(0),A);
            GO TO NEXT;
            END;
          /*

          ON ENCOUNTERING AN END OF FILE ON OLD, WHICH INDICATES
          THAT NO MORE RECORDS EXIST IN THE MAIN FILE, SET BIT TO
          A '1'B.  PRINT THE SOCIAL SECURITY NUMBERS OF THE
          REMAINING RECORDS TO BE UPDATED WITH THE MESSAGE
          'MISSING FROM MAIN FILE.'
                                                                   */
          ON ENDFILE(OLD) BEGIN;
            IF BIT = '1'B THEN GO TO FINISH;
            BIT = '1'B;
NOTE:       PUT FILE(SYSPRINT) EDIT('MISSING FROM MAIN FILE')
              (COL(26),A);
ADVANCE:    GET FILE(SYSIN) EDIT(ITEM)(SKIP,F(2));
            IF ITEM = 99 THEN DO;
              GET FILE(SYSIN) EDIT(SS_NO)(SKIP,F(2));
              PUT FILE(SYSPRINT) EDIT(SS_NO)(SKIP(2),X(4),A);
              GO TO NOTE;
              END;
            GO TO ADVANCE;
            END;
          /*

          READ THE CURRENT DATE.
                                                                   */
          GET FILE(SYSIN) EDIT(CUR_DATE)(COL(1),A(8));
          /*

          PRINT THE HEADER INFORMATION FOR THE FIRST PAGE OF
          PRINT.
                                                                   */
          PUT FILE(SYSPRINT) EDIT(CUR_DATE)(PAGE,A);
          PUT FILE(SYSPRINT) EDIT('UPDATE',
            'OUTPUT DESCRIPTION: A LISTING BY SOCIAL SECURITY',
            ' NUMBER OF THE OLD AND NEW CONTENTS OF ALL ITEMS',
            ' WHICH HAVE BEEN UPDATED')(SKIP,A,SKIP,3 A);
          PUT FILE(SYSPRINT) EDIT('SOCIAL SECURITY NO','ITEM',
            'PREVIOUS CONTENTS','PRESENT CONTENTS')
              (SKIP(6),A,COL(25),A,COL(35),A,COL(85),A);
          PUT FILE(SYSPRINT) EDIT('__________________','____',
            '_________________________','____________________')
            (SKIP(0),COL(1),A,COL(25),A,COL(35),A,COL(85),A);
          /*

          READ AND PRINT THE SOCIAL SECURITY NUMBER OF THE NEXT
          RECORD TO BE UPDATED.
                                                                   */
FOLLOW:   GET FILE(SYSIN) EDIT(SS_NO)(COL(1),A(9));
          PUT FILE(SYSPRINT) EDIT(SS_NO)(SKIP(2),X(4),A);
```

```
        /*

        READ A RECORD FROM THE MAIN FILE.
                                                                   */
SEARCH: READ FILE(OLD) INTO(PERSONAL);
        /*

        IF THE RECORD TO BE UPDATED HAS NOT YET BEEN FOUND IN
        THE MAIN FILE, COPY THE NEW RECORD ONTO THE NEW MAIN
        FILE.
                                                                   */
CHECK:  IF PERSONAL.SOC_SEC_NO < SS_NO THEN DO;
          WRITE FILE(NEW) FROM(PERSONAL);
          GO TO SEARCH;
          END;
        /*

        IF THE RECORD TO BE UPDATED DOES NOT EXIST IN THE MAIN
        FILE, PRINT THE SOCIAL SECURITY NUMBER WITH THE MESSAGE
        'MISSING FROM MAIN FILE.'  READ THE REMAINING CARDS IN
        THIS RECORD.  THEN, READ AND PRINT THE SOCIAL SECURITY
        NUMBER OF THE NEXT RECORD TO BE UPDATED.
                                                                   */
        IF PERSONAL.SOC_SEC_NO > SS_NO THEN DO;
          PUT FILE(SYSPRINT) EDIT('MISSING FROM MAIN FILE')
            (COL(26),A);
PASS:     GET FILE(SYSIN) EDIT(ITEM)(SKIP,F(2));
          IF ITEM ¬= 99 THEN GO TO PASS;
            ELSE DO;
              GET FILE(SYSIN) EDIT(SS_NO)(COL(1),A(9));
              PUT FILE(SYSPRINT) EDIT(SS_NO)(SKIP(2),X(4),A);
              GO TO CHECK;
              END;
          END;
        /*

        READ THE UPDATE NUMBER OF THE NEXT ITEM TO BE UPDATED.
                                                                   */
NEXT:   GET FILE(SYSIN) EDIT(ITEM)(SKIP,F(2));
        PUT SKIP(2);
        /*

        IF ITEM IS EQUAL TO NINETY-NINE, THE END OF THE UPDATES
        FOR A STUDENT RECORD HAS BEEN REACHED.  COPY THE
        UPDATED RECORD ONTO THE NEW MAIN FILE.
                                                                   */
        IF ITEM = 99 THEN DO;
          WRITE FILE(NEW) FROM(PERSONAL);
          GO TO FOLLOW;
          END;
        /*

        SET LOC AND TRANSFER TO THE CORRECT UPDATE ROUTINE.
                                                                   */
        LOC = TRANSFER(ITEM);
        GO TO LOC;
        /*

        UPDATE THE SOCIAL SECURITY NUMBER BY CALLING LETTER.
                                                                   */
NO_1:   CALL LETTER(SOC_SEC_NO,9);
        /*

        UPDATE THE COMPLETE NAME.  PRINT THE PREVIOUS AND
        PRESENT CONTENTS OF NAME.
                                                                   */
NO_2:   PUT FILE(SYSPRINT) EDIT(ITEM,NAME)(COL(26),F(2),X(7),3 (
          A,X(2)));
        GET FILE(SYSIN) EDIT(NAME)(X(3),A(14),X(1),A(10),X(5),
          A(14));
        PUT FILE(SYSPRINT) EDIT(NAME)(X(8),3 (A,X(2)));
        GO TO NEXT;
        /*

        UPDATE THE LAST NAME BY CALLING LETTER.
                                                                   */
NO_3:   CALL LETTER(LAST,14);
        /*

        UPDATE THE FIRST NAME BY CALLING LETTER.
                                                                   */
NO_4:   CALL LETTER(FIRST,10);
        /*
```

```
         UPDATE THE MIDDLE NAME BY CALLING LETTER.
                                                                 */
NO_5:    CALL LETTER(MIDDLE,14);
         /*

         UPDATE THE COMPLETE PERMANENT ADDRESS.  PRINT THE
         PREVIOUS AND PRESENT CONTENTS OF PERMANENT ADDRESS.
                                                                 */
NO_6:    PUT FILE(SYSPRINT) EDIT(ITEM,PERM_ADDR.STREET,
           PERM_ADDR.CITY)(COL(26),F(2),X(8),2(A,X(2)));
         STORE_C = PERM_ADDR.STATE || '   ' || PERM_ADDR.ZIP;
         GET FILE(SYSIN) EDIT(PERM_ADDR)(X(3),A(20),X(5),A(20),
           X(5),A(2),X(3),A(5));
         PUT FILE(SYSPRINT) EDIT(PERM_ADDR.STREET,PERM_ADDR.CITY)
           (X(7),2 (A,X(2)));
         PUT FILE(SYSPRINT) EDIT(STORE_C,PERM_ADDR.STATE,
           PERM_ADDR.ZIP)(COL(35),A,X(41),2 (A,X(2)));
         GO TO NEXT;
         /*

         UPDATE THE STREET IN PERMANENT ADDRESS BY CALLING
         LETTER.
                                                                 */
NO_7:    CALL LETTER(PERM_ADDR.STREET,20);
         /*

         UPDATE THE CITY IN PERMANENT ADDRESS BY CALLING LETTER,
                                                                 */
NC_8:    CALL LETTER(PERM_ADDR.CITY,20);
         /*

         UPDATE THE STATE IN PERMANENT ADDRESS BY CALLING LETTER.
                                                                 */
NO_9:    CALL LETTER(PERM_ADDR.STATE,2);
         /*

         UPDATE THE ZIP CODE IN PERMANENT ADDRESS BY CALLING
         LETTER.
                                                                 */
NO_10:   CALL LETTER(PERM_ADDR.ZIP,5);
         /*

         UPDATE THE PERMANENT PHONE NUMBER BY CALLING LETTER.
                                                                 */
NO_11:   CALL LETTER(PERM_PHONE,I0);
         /*

         UPDATE THE COMPLETE LOCAL ADDRESS.  PRINT THE PREVIOUS
         AND PRESENT CONTENTS OF LOCAL ADDRESS.
                                                                 */
NO_I2:   PUT FILE(SYSPRINT) EDIT(ITEM,LOCAL_ADDR,STREET,
           LOCAL_ADDR.CITY)(COL(26),F(2),X(8),2(A,X(2)));
         STORE_C = LOCAL_ADDR.STATE || '   ' || LOCAL_ADDR.ZIP;
         GET FILE(SYSIN) EDIT(LOCAL_ADDR)(X(3),A(20),X(5),A(20),
           X(5),A(2),X(3),A(5));
         PUT FILE(SYSPRINT) EDIT(LOCAL_ADDR.STREET,
           LOCAL_ADDR.CITY)(X(7),2 (A,X(2)));
         PUT FILE(SYSPRINT) EDIT(STORE_C,LOCAL_ADDR.STATE,
           LOCAL_ADDR.ZIP)(COL(35),A,X(41),2 (A,X(2)));
         GO TC NEXT;
         /*

         UPDATE THE STREET IN LOCAL ADDRESS BY CALLING LETTER.
                                                                 */
NO_13:   CALL LETTER(LOCAL_ADDR,STREET,20);
         /*

         UPDATE THE CITY IN LOCAL ADDRESS BY CALLING LETTER.
                                                                 */
NO_14:   CALL LETTER(LOCAL_ADDR;CITY,20);
         /*

         UPDATE THE STATE IN LOCAL ADDRESS BY CALLING LETTER.
                                                                 */
NO_I5:   CALL LETTER(LOCAL_ADDR)STATE,2);
         /*

         UPDATE THE ZIP CODE IN LOCAL ADDRESS BY CALLING LETTER.
                                                                 */
NO_I6:   CALL LETTER(LOCAL_ADDR.ZIP,5);
         /*

         UPDATE THE LOCAL PHONE NUMBER BY CALLING LETTER,
                                                                 */
```

```
NO_17:  CALL LETTER(LOCAL_PHONE,7);
        /*

        UPDATE SEX BY CALLING LETTER.
                                                              */
NO_18:  CALL LETTER(SEX,1);
        /*

        UPDATE THE COMPLETE BIRTH DATE.  PRINT THE PREVIOUS AND
        PRESENT CONTENTS OF BIRTH DATE.
                                                              */
NO_19:  PUT FILE(SYSPRINT) EDIT(ITEM,BIRTH_DATE)(COL(26),F(2),
          X(7),3 (F(2),X(2)));
        GET FILE(SYSIN) EDIT(BIRTH_DATE)(X(3),3 (F(2),X(3)));
        PUT FILE(SYSPRINT) EDIT(BIRTH_DATE)(X(40),3 (F(2),X(2)))
          ;
        GO TO NEXT;
        /*

        UPDATE THE MONTH IN BIRTH DATE BY CALLING NUMBER.
                                                              */
NO_20:  CALL NUMBER(BIRTH_DATE.MONTH,2);
        /*

        UPDATE THE DAY IN BIRTH DATE BY CALLING NUMBER.
                                                              */
NO_21:  CALL NUMBER(BIRTH_DATE.DAY,2);
        /*

        UPDATE THE YEAR IN BIRTH DATE BY CALLING NUMBER.
                                                              */
NO_22:  CALL NUMBER(BIRTH_DATE.YR,2);
        /*

        UPDATE THE MARITAL STATUS BY CALLING NUMBER.
                                                              */
NO_23:  CALL NUMBER(MARITAL_STAT,1);
        /*

        UPDATE THE NUMBER OF DEPENDENT CHILDREN BY CALLING
        NUMBER.
                                                              */
NO_24:  CALL NUMBER(DEP_CHILDREN,1);
        /*

        UPDATE THE AGE RANGE OF DEPENDENT CHILDREN BY CALLING
        ARRAY.
                                                              */
NO_25:  CALL ARRAY(2,AGE_RANGE,2);
        /*

        UPDATE ALL OF THE WORK EXPERIENCES BY CALLING ARRAY.
                                                              */
NO_26:  CALL ARRAY(6,WORK_EXP,2);
        /*

        UPDATE THE FIRST BLANK WORK EXPERIENCE BY CALLING ARRAY.
        PRINT AN ERROR MESSAGE IF THE ARRAY IS FULL.
                                                              */
NO_27:  DO I = 1 TO 6 WHILE(WORK_EXP(I) ¬= 0);
          END;
        IF I = 7 THEN DO;
          PUT FILE(SYSPRINT) EDIT(ITEM,'MORE THAN SIX WORK ',
             'EXPERIENCES')(COL(26),F(2),COL(85),2 A);
          GO TO NEXT;
          END;
        CALL NUMBER(WORK_EXP(I),2);
        /*

        UPDATE ALL OF THE TRAVEL EXPERIENCES BY CALLING ARRAY.
                                                              */
NO_28:  CALL ARRAY(6,TRAVEL_EXP,2);
        /*

        UPDATE THE FIRST BLANK TRAVEL EXPERIENCE BY CALLING
        ARRAY.  PRINT AN ERROR MEASAGE IF THE ARRAY IS FULL.
                                                              */
NO_29:  DO I = 1 TO 6 WHILE(TRAVEL_EXP(I) ¬= 0);
          END;
        IF I = 7 THEN DO;
          PUT FILE(SYSPRINT) EDIT(ITEM,'MORE THAN SIX TRAVEL ',
             'EXPERIENCES')(COL(26),F(2),COL(85),2 A);
          GO TO NEXT;
          END;
        CALL NUMBER(TRAVEL_EXP(I),2);
        /*

        UPDATE ALL INFORMATION IN CURRENT EMPLOYMENT.  PRINT THE
        PREVIOUS AND PRESENT CONTENTS OF CURRENT EMPLOYMENT.
                                                                 */
NO_30:  PUT FILE(SYSPRINT) EDIT(ITEM,CUR_EMPLOYMENT)(COL(26),
          F(2),X(7),A,X(2),F(2));
        GET FILE(SYSIN) EDIT(CUR_EMPLOYMENT)(X(3),A(20),X(5),
          F(2));
        PUT FILE(SYSPRINT) EDIT(CUR_EMPLOYMENT)(COL(85),A,X(2),
          F(2));
        GO TO NEXT;
        /*

        UPDATE THE NAME OF THE EMPLOYER IN CURRENT EMPLOYMENT
        BY CALLING LETTER.
                                                                 */
NO_31:  CALL LETTER(CUR_EMPLOYMENT.EMPLOYER,20);
        /*

        UPDATE THE HOURS WORKED PER WEEK IN CURRENT EMPLOYMENT
        BY CALLING NUMBER.
                                                                 */
NO_32:  CALL NUMBER(CUR_EMPLOYMENT.HRS_PER_WK,2);
        /*

        UPDATE THE INFORMATION ON ONE OF THE COLLEGES ATTENDED.
        PRINT THE PREVIOUS AND PRESENT INFORMATION ABOUT THIS
        COLLEGE.
                                                                 */
NO_33:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        PUT FILE(SYSPRINT) EDIT(ITEM,COL_ATT(INDEX))(COL(26),
          F(2),X(7),3 (A,X(2)));
        GET FILE(SYSIN) EDIT(COL_ATT(INDEX))(X(3),A(30),X(5),
          2 (A(6),X(4)));
        PUT FILE(SYSPRINT) EDIT(COL_ATT(INDEX))(X(4),3 (A,X(2)))
        ;
        GO TO NEXT;
        /*

        UPDATE THE INSTITUTION NAME FOR ONE OF THE COLLEGES
        ATTENDED BY CALLING LETTER.
                                                                 */
NO_34:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL LETTER(INSTITUTION(INDEX),30);
        /*

        UPDATE THE DATES OF ATTENDANCE FOR ONE OF THE COLLEGES
        ATTENDED.  PRINT THE PREVIOUS AND PRESENT INFORMATION
        ABOUT THE DATES OF ATTENDANCE.
                                                                 */
NO_35:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        PUT FILE(SYSPRINT) EDIT(ITEM,DATES(INDEX))(COL(26),F(2),
          X(7),2 (A,X(2)));
        GET FILE(SYSIN) EDIT(DATES(INDEX))(X(3),2 (A(6),X(4)));
        PUT FILE(SYSPRINT) EDIT(DATES(INDEX))(X(36),2 (A(6),
          X(4)));
        GO TO NEXT;
        /*

        UPDATE THE BEGINNING DATE OF ATTENDANCE FOR ONE OF THE
        COLLEGES ATTENDED BY CALLING LETTER.
                                                                 */
NO_36:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL LETTER(BEGIN(INDEX),6);
        /*

        UPDATE THE ENDING DATE OF ATTENDANCE FOR ONE OF THE
        COLLEGES ATTENDED BY CALLING LETTER.
                                                                 */
NO_37:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL LETTER(END(INDEX),6);
        /*

        UPDATE THE GRADE POINT AVERAGE BY CALLING NUMBER.
                                                                 */
NO_38:  CALL NUMBER(GPA,1);
        /*

        UPDATE THE MYERS BRIGGS TYPE INDICATOR BY CALLING
        NUMBER.
                                                                 */
NO_39:  CALL NUMBER(MYERS_BRIGGS,2);
```

```
        /*

        UPDATE THE ENTIRE CEP ENTRANCE DATE.  PRINT THE PREVIOUS
        AND PRESENT INFORMATION ABOUT THE CEP ENTRANCE DATE.
                                                                  */
NO_40:  PUT FILE(SYSPRINT) EDIT(ITEM,CEP_ENTRANCE)(COL(26),F(2),
          X(7),2 (F(2),X(2)),F(5,1));
        GET FILE(SYSIN) EDIT(CEP_ENTRANCE)(X(3),2 (F(2),X(3)),
          F(4,1));
        PUT FILE(SYSPRINT) EDIT(CEP_ENTRANCE)(X(38),2 (F(2),X(2)
          ),F(5,1));
        GO TO NEXT;
        /*

        UPDATE THE YEAR IN THE CEP ENTRANCE DATE BY CALLING
        NUMBER.
                                                                  */
NO_41:  CALL NUMBER(CEP_ENTRANCE.YR,2);
        /*

        UPDATE THE QUARTER IN THE CEP ENTRANCE DATE BY CALLING
        NUMBER.
                                                                  */
NO_42:  CALL NUMBER(CEP_ENTRANCE.QTR,1);
        /*

        UPDATE THE HOURS COMPLETED UPON ENTRANCE INTO CEP.
        PRINT THE PREVIOUS AND PRESENT CONTENTS OF HOURS
        COMPLETED.
                                                                  */
NO_43:  PUT FILE(SYSPRINT) EDIT(ITEM,HRS_COM)(COL(26),F(2),X(7),
          F(5,1));
        GET FILE(SYSIN) EDIT(HRS_COM)(X(3),F(5,1));
        PUT FILE(SYSPRINT) EDIT(HRS_COM)(COL(85),F(5,1));
        GO TO NEXT;
        /*

        UPDATE THE AREA OF SPECIALIZATION BY CALLING NUMBER.
                                                                  */
NO_44:  CALL NUMBER(SPECIALIZATION,1);
        /*

        UPDATE THE TEAM NUMBER BY CALLING NUMBER.
                                                                  */
NO_45:  CALL NUMBER(TEAM,1);
        /*

        UPDATE THE SEMINAR NUMBER BY CALLING NUMBER.
                                                                  */
NO_46:  CALL NUMBER(SEMINAR,2);
        /*

        UPDATE THE ENTIRE EXPECTED DATE OF GRADUATION.  PRINT
        THE PREVIOUS AND PRESENT CONTENTS OF THE EXPECTED
        DATE OF GRADUATION.
                                                                  */
NO_47:  PUT FILE(SYSPRINT) EDIT(ITEM,EXP_GRAD_DATE)(COL(26),F(2)
          ,X(7),2 (F(2),X(2)));
        GET FILE(SYSIN) EDIT(EXP_GRAD_DATE)(X(3),2 (F(2),X(3)));
        PUT FILE(SYSPRINT) EDIT(EXP_GRAD_DATE)(X(43),2 (F(2),
          X(2)));
        GO TO NEXT;
        /*

        UPDATE THE YEAR IN THE EXPECTED DATE OF GRADUATION BY
        CALLING NUMBER.
                                                                  */
NO_48:  CALL NUMBER(EXP_GRAD_DATE.YR,2);
        /*

        UPDATE THE QUARTER IN THE EXPECTED DATE OF GRADUATION BY
        CALLING NUMBER.
                                                                  */
NO_49:  CALL NUMBER(EXP_GRAD_DATE.QTR,1);
        /*

        UPDATE THE GRADE FOR ONE OF THE REQUIRED COURSES BY
        CALLING LETTER.
                                                                  */
NO_50:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL LETTER(REQ_COURSE_GRADE(INDEX),1);
        /*

        UPDATE ALL INFORMATION ON ONE REGISTRATION.  PRINT THE
        PREVIOUS AND PRESENT INFORMATION ABOUT THIS
        REGISTRATION.
                                                                  */
NO_51:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        PUT FILE(SYSPRINT) EDIT(ITEM,REGISTRATION.DATE(INDEX))
          (COL(26),F(2),X(7),2 (F(2),X(2)));
        DO I = 1 TO 5;
          STORE_S(I) = COURSES(INDEX,I);
          END;
        GET FILE(SYSIN) EDIT(REGISTRATION(INDEX))(X(3),F(2),
          X(3),F(1),X(4),3 (A(6),X(4),F(2),X(3),A(1),X(4)),
        COL(6),2 (A(6),X(4),F(2),X(3),A(1),X(4)));
        PUT FILE(SYSPRINT) EDIT(REGISTRATION.DATE(INDEX))(X(44),
          2 (F(2),X(2)));
        PUT FILE(SYSPRINT) EDIT((STORE_S(I) DO I = 1 TO 3),
          (COURSES(INDEX,I) DO I = 1 TO 3),(STORE_S(I) DO I = 4
          TO 5),(COURSES(INDEX,I) DO I = 4 TO 5))
          (COL(35),3 (A,X(2),F(2),X(2),A,X(2)),COL(85),3 (A,X(2)
          ,F(2),X(2),A,X(2)),COL(35),2 (A,X(2),F(2),X(2),A,X(2))
          ,COL(85),2 (A,X(1),F(2),X(2),A,X(2)));
        GO TO NEXT;
        /*

        UPDATE THE ENTIRE DATE FOR ONE OF THE REGISTRATIONS BY
        CALLING TIME.
                                                                  */
NO_52:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL TIME(REGISTRATION.DATE(INDEX));
        /*

        UPDATE THE YEAR FOR ONE OF THE REGISTRATIONS BY CALLING
        NUMBER.
                                                                  */
NO_53:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL NUMBER(REGISTRATION.YR(INDEX),2);
        /*

        UPDATE THE QUARTER FOR ONE OF THE REGISTRATIONS BY
        CALLING NUMBER.
                                                                  */
NO_54:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL NUMBER(REGISTRATION.QTR(INDEX),1);
        /*

        UPDATE ALL INFORMATION ON THE COURSES FOR ONE
        REGISTRATION.  PRINT THE PREVIOUS AND PRESENT
        INFORMATION ABOUT THESE COURSES.
                                                                  */
NO_55:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        PUT FILE(SYSPRINT) EDIT(ITEM,(COURSES(INDEX,I) DO I = 1
          TO 3))(COL(26),F(2),X(7),3 (A,X(2),F(2),X(2),A,X(2)));
        DO I = 1 TO 5;
          STORE_S(I) = COURSES(INDEX,I);
          END;
        GET FILE(SYSIN) EDIT((COURSES(INDEX,I) DO I = 1 TO 5))
          (X(3),3 (A(6),X(4),F(2),X(3),A(1),X(4)),COL(5),2 (A(6)
          ,X(4),F(2),X(3),A(1)));
        PUT FILE(SYSPRINT) EDIT((COURSES(INDEX,I) DO I = I TO 3)
          )(COL(85),3 (A,X(2),F(2),X(2),A,X(3)));
        PUT FILE(SYSPRINT) EDIT((STORE_S(I) DO I = 3 TO 4),
          (COURSES(INDEX,I) DO I = 3 TO 4))(COL(35),2 (A,X(2),
         F(2),X(2),A,X(2)),COL(85),2 (A,X(2),F(2),X(2),A,X(2)));
        GO TO NEXT;
        /*

        UPDATE THE COURSE NUMBER FOR A COURSE FOR ONE OF THE
        REGISTRATIONS BY CALLING LETTER.
                                                                  */
NO_56:  GET FILE(SYSIN) EDIT(INDEX,INDEX_1)(X(3),F(2));
        CALL LETTER(COURSE_NO(INDEX,INDEX_1),6);
        /*

        UPDATE THE CREDIT HOURS FOR A COURSE FOR ONE OF THE
        REGISTRATIONS BY CALLING NUMBER.
                                                                  */
NO_57:  GET FILE(SYSIN) EDIT(INDEX,INDEX_1)(X(3),F(2));
        CALL NUMBER(HRS(INDEX,INDEX_1),2);
        /*

        UPDATE THE GRADE FOR A COURSE FOR ONE OF THE
        REGISTRATIONS BY CALLING LETTER.
                                                                  */
NO_58:  GET FILE(SYSIN) EDIT(INDEX,INDEX_1)(X(3),F(2));
        CALL LETTER(REGISTRATION.GRADE(INDEX,INDEX_1),1);
        /*

        UPDATE ALL OF THE INFORMATION ON ONE LEARNING ACTIVITY.
        PRINT THE PREVIOUS AND PRESENT INFORMATION ABOUT THIS
        ACTIVITY.
                                                                   */
NO_59:  GET FILE(SYSIN) EDIT(AREA#,ACTIVITY#)(2 (X(3),F(2)));
        INDEX = ENTRY(AREA#) + ACTIVITY#;
        PUT FILE(SYSPRINT) EDIT(ITEM,LEARNING_A(INDEX))
          (COL(26),F(2),X(7),F(2),X(2),A,X(2),6 (F(2),X(2)),
          F(3),X(2),2(B,X(2)));
        GET FILE(SYSIN) EDIT(LEARNING_A(INDEX))(X(3),F(2),X(3),
          A(6),X(4),6 (F(2),X(3)),F(3),X(2),2 (B(1),X(4)));
        PUT FILE(SYSPRINT) EDIT(LEARNING_A(INDEX))(X(5),
          F(2),X(2),A,X(2),6 (F(2),X(2)),F(3),X(2),2 (B,X(2)));
        GO TO NEXT;
        /*

        UPDATE THE FACULTY MEMBER FOR ONE OF THE LEARNING
        ACTIVITIES BY CALLING NUMBER.
                                                                   */
NO_60:  GET FILE(SYSIN) EDIT(AREA#,ACTIVITY#)(2 (X(3),F(2)));
        INDEX = ENTRY(AREA#) + ACTIVITY#;
        CALL NUMBER(FACULTY(INDEX),2);
        /*

        UPDATE THE DATE FOR ONE OF THE LEARNING ACTIVITIES BY
        LETTER.
                                                                   */
NO_61:  GET FILE(SYSIN) EDIT(AREA#,ACTIVITY#)(2 (X(3),F(2)));
        INDEX = ENTRY(AREA#) + ACTIVITY#;
        CALL LETTER(LEARNING_A.DATE(INDEX),6);
        /*

        UPDATE THE COMMENTS FOR ONE OF THE LEARNING ACTIVITIES
        BY CALLING ARRAY.
                                                                   */
NO_62:  GET FILE(SYSIN) EDIT(AREA#,ACTIVITY#)(2 (X(3),F(2)));
        INDEX = ENTRY(AREA#) + ACTIVITY#;
        CALL ARRAY(6,COMMENT(INDEX,*),2);
        /*

        UPDATE THE ACTIVITY TITLE FOR ONE OF THE LEARNING
        ACTIVITIES BY CALLING NUMBER.
                                                                   */
NO_63:  GET FILE(SYSIN) EDIT(AREA#,ACTIVITY#)(2 (X(3),F(2)));
        INDEX = ENTRY(AREA#) + ACTIVITY#;
        CALL NUMBER(ACTIVITY_TITLE(INDEX),3);
        /*

        UPDATE THE GROUP STATUS FOR ONE OF THE LEARNING
        ACTIVITIES BY CALLING BITS.
                                                                   */
NO_64:  GET FILE(SYSIN) EDIT(AREA#,ACTIVITY#)(2 (X(3),F(2)));
        INDEX = ENTRY(AREA#) + ACTIVITY#;
        CALL BITS(GROUP(INDEX));
        /*

        UPDATE THE SUBSTITUTE STATUS FOR ONE OF THE LEARNING
        ACTIVITIES BY CALLING BITS.
                                                                   */
NO_65:  GET FILE(SYSIN) EDIT(AREA#,ACTIVITY#)(2 (X(3),F(2)));
        INDEX = ENTRY(AREA#) + ACTIVITY#;
        CALL BITS(SUBSTITUTE(INDEX));
        /*

        UPDATE ALL OF THE INFORMATION ON ONE FIELD EXPERIENCE.
        PRINT THE PREVIOUS AND PRESENT INFORMATION ABOUT THIS
        FIELD EXPERIENCE.
                                                                   */
NO_66:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        PUT FILE(SYSPRINT) EDIT(ITEM,SCHOOL(INDEX),FIELD.GRADE
          (INDEX),FIELD.LEVEL(INDEX),FIELD.DATE(INDEX),CLASSROOM
          (INDEX),TEACHER(INDEX))(COL(26),F(2),X(7),F(2),X(2),
          F(2),X(2),F(1),X(2),F(2),X(2),2 (F(2),X(2)),A);
        DO I = 1 TO 6;
          STORE_A(I) = F_ROLE(INDEX,I);
          STORE_A(I + 6) = STRENGTHS(INDEX,I);
          STORE_A(I + 12) = WEAKNESSES(INDEX,I);
          END;
        GET FILE(SYSIN) EDIT(FIELD(INDEX))(X(3),2 (F(2),X(3)),
          F(1),X(4),F(2),X(3),2 (F(1),X(4)),A(14),2 (COL(6),12 (
          F(2),X(3))));
        PUT FILE(SYSPRINT) EDIT(SCHOOL(INDEX),FIELD.GRADE(INDEX)
           ,FIELD.LEVEL(INDEX),FIELD.DATE(INDEX),CLASSROOM(INDEX)
           ,TEACHER(INDEX))(X(13),2 (F(2),X(2)),F(1),X(2),F(2),
           X(2),F(1),X(2),F(2),X(2),A);
         PUT FILE(SYSPRINT) EDIT((STORE_A(I) DO I = 1 TO 12),
           F_ROLE(INDEX,*),STRENGTHS(INDEX,*),(STORE_A(I) DO I =
           13 TO 18),WEAKNESSES(INDEX,*))(COL(35),12 (F(2),X(2)),
           X(1),12 (F(2),X(2)),COL(35),6 (F(2),X(2)),X(25),6(F(2)
           ,X(2)));
         GO TO NEXT;
         /*

         UPDATE THE SCHOOL FOR ONE OF THE FIELD EXPERIENCES BY
         CALLING NUMBER.
                                                                    */
NO_67:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL NUMBER(SCHOOL(INDEX),2);
         /*

         UPDATE THE GRADE FOR ONE OF THE FIELD EXPERIENCES BY
         CALLING NUMBER.
                                                                    */
NO_68:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL NUMBER(FIELD.GRADE(INDEX),2);
         /*

         UPDATE THE LEVEL IN CER FOR ONE OF THE FIELD EXPERIENCES
         BY CALLING NUMBER.
                                                                    */
NO_69:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL NUMBER(FIELD.LEVEL(INDEX),1);
         /*

         UPDATE THE ENTIRE DATE FOR ONE OF THE FIELD EXPERIENCES
         BY CALLING TIME.
                                                                    */
NO_70:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL TIME(FIELD.DATE(INDEX));
         /*

         UPDATE THE YEAR FOR ONE OF THE FIELD EXPERIENCES BY
         CALLING NUMBER.
                                                                    */
NO_71:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL NUMBER(FIELD.YR(INDEX),2);
         /*

         UPDATE THE QUARTER FOR ONE OF THE FIELD EXPERIENCES BY
         CALLING NUMBER.
                                                                    */
NO_72:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL NUMBER(FIELD.QTR(INDEX),1);
         /*

         UPDATE THE CLASSROOM TYPE FOR ONE OF THE FIELD
         EXPERIENCES BY CALLING NUMBER.
                                                                    */
NO_73:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL NUMBER(CLASSROOM(INDEX),2);
         /*

         UPDATE THE SUPERVISING TEACHER FOR ONE OF THE FIELD
         EXPERIENCES BY CALLING LETTER.
                                                                    */
NO_74:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL LETTER(TEACHER(INDEX),14);
         /*

         UPDATE THE ROLES FOR ONE OF THE FIELD EXPERIENCES BY
         CALLING ARRAY.
                                                                    */
NO_75:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL ARRAY(6,F_ROLE(INDEX,*),2);
         /*

         UPDATE THE STRENGTHS FOR ONE OF THE FIELD EXPERIENCES BY
         CALLING ARRAY.
                                                                    */
NO_76:   GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
         CALL ARRAY(6,STRENGTHS(INDEX,*),2);
         /*

         UPDATE THE NEEDS FOR ONE OF THE FIELD EXPERIENCES BY
         CALLING ARRAY.
                                                                    */
```

```
NO_77:  GET FILE(SYSIN) EDIT(INDEX)(X(3),F(2));
        CALL ARRAY(6,WEAKNESSES(INDEX,*),2);
        /*

        UPDATE ALL OF THE INFORMATION ON THE COMMUNITY
        EXPERIENCE.  PRINT THE PREVIOUS AND PRESENT INFORMATION
        ABOUT THIS EXPERIENCE.
                                                              */
NO_78:  PUT FILE(SYSPRINT) EDIT(ITEM,AGENCY,AGENCY_REP,
          COMMUNITY.LEVEL,COMMUNITY.DATE)(COL(26),F(2),X(7),2 (
          A,X(2)),F(1),X(2),F(2),X(2),F(1));
        DO I = 1 TO 4;
          STORE_A(I) = C_ROLE(I);
          STORE_A(I + 4) = C_COMMENT(I);
          END;
        GET FILE(SYSIN) EDIT(COMMUNITY)(X(3),A(20),X(5),A(14),
          X(1),F(1),X(4),F(2),X(3),F(1),COL(6),8 (F(2),X(3)));
        PUT FILE(SYSPRINT) EDIT(AGENCY,AGENCY_REP,
          COMMUNITY.LEVEL,COMMUNITY.DATE)(X(4),2 (A,X(2)),F(1),
          X(2),F(2),X(2),F(1));
        PUT FILE(SYSPRINT) EDIT((STORE_A(I) DO I = 1 TO 8),
          C_ROLE,C_COMMENT)(COL(35),8 (F(2),X(2)),X(20),8 (F(2)
          ,X(2)));
        GO TO NEXT;
        /*

        UPDATE THE AGENCY FOR THE COMMUNITY EXPERIENCE BY
        CALLING LETTER.
                                                              */
NO_79:  CALL LETTER(AGENCY,20);
        /*

        UPDATE THE AGENCY REPRESENTATIVE FOR THE COMMUNITY
        EXPERIENCE BY CALLING LETTER.
                                                              */
NO_80:  CALL LETTER(AGENCY_REP,14);
        /*

        UPDATE THE LEVEL IN CEP FOR THE COMMUNITY EXPERIENCE BY
        CALLING NUMBER.
                                                              */
NO_81:  CALL NUMBER(COMMUNITY.LEVEL,1);
        /*

        UPDATE THE ENTIRE DATE FOR THE COMMUNITY EXPERIENCE BY
        CALLING TIME.
                                                              */
NO_82:  CALL TIME(COMMUNITY.DATE);
        /*

        UPDATE THE YEAR OF THE COMMUNITY EXPERIENCE BY CALLING
        NUMBER.
                                                              */
NO_83:  CALL NUMBER(COMMUNITY.YR,2);
        /*

        UPDATE THE QUARTER OF THE COMMUNITY EXPERIENCE BY
        CALLING NUMBER.
                                                              */
NO_84:  CALL NUMBER(COMMUNITY.QTR,1);
        /*

        UPDATE THE ROLES FOR THE COMMUNITY EXPERIENCE BY
        CALLING ARRAY.
                                                              */
NO_85:  CALL ARRAY(4,C_ROLE,2);
        /*

        UPDATE THE COMMENTS ABOUT THE COMMUNITY EXPERIENCE BY
        CALLING ARRAY.
                                                              */
NO_86:  CALL ARRAY(4,C_COMMENT,2);
        /*

        ARRAY UPDATES THE ELEMENTS OF AN ARRAY.  IT PRINTS THE
        PREVIOUS AND PRESENT CONTENTS OF THE ARRAY.  THE
        ARGUMENTS PASSED TO ARRAY ARE THE NUMBER OF ELEMENTS
        IN THE ARRAY, THE NAME OF THE ARRAY TO BE PROCESSED AND
        THE NUMBER OF DIGITS OF EACH ELEMENT.IN THE ARRAY.
                                                              */
ARRAY:  PROCEDURE(N_REP,VALUES,LEN);
        DECLARE VALUES(*) FIXED BINARY;
        PUT FILE(SYSPRINT) EDIT(ITEM,(VALUES(I) DO I = 1 TO
          N_REP))(COL(26),F(2),X(7),(N_REP)(F(LEN),X(2)));
```

```
         GET FILE(SYSIN) EDIT((VALUES(I) DO I = 1 TO N_REP))(X(3)
           ,(N_REP)(F(LEN),X(5 - LEN)));
         PUT FILE(SYSPRINT) EDIT((VALUES(I) DO I = 1 TO N_REP))
           (COL(85),(N_REP)(F(LEN),X(2)));
         GO TO NEXT;
         END;
         /*

         BITS UPDATES THE CONTENTS OF A BIT.  IT PRINTS THE
         PREVIOUS AND PRESENT CONTENTS OF THE BIT.  THE ARGUMENT
         PASSED TO BIT IS THE NAME OF THE BIT.
                                                               */
BITS:    PROCEDURE(BIT);
         DECLARE BIT BIT(1);
         PUT FILE(SYSPRINT) EDIT(ITEM,BIT)(COL(26),F(2),X(7),
           B(1));
         GET FILE(SYSIN) EDIT(BIT)(X(3),B(1));
         PUT FILE(SYSPRINT) EDIT(BIT)(COL(85),B(1));
         GO TO NEXT;
         END;
         /*

         LETTER UPDATES THE CONTENTS OF A CHARACTER STRING. IT
         PRINTS THE PREVIOUS AND PRESENT CONTENTS OF THE
         CHARACTER STRING.  THE ARGUMENTS PASSED TO LETTER ARE
         THE NAME AND THE LENGTH OF THE CHARACTER STRING.
                                                               */
LETTER:  PROCEDURE(LET,LEN);
         DECLARE LET CHARACTER(*);
         PUT FILE(SYSPRINT) EDIT(ITEM)(COL(26),F(2));
         PUT FILE(SYSPRINT) EDIT(LET)(X(7),A(LEN));
         GET FILE(SYSIN) EDIT(LET)(X(3),A(LEN));
         PUT FILE(SYSPRINT) EDIT(LET)(COL(85),A(LEN));
         GO TO NEXT;
         END;
         /*

         NUMBER UPDATES THE CONTENTS OF AN ARITHMETIC VALUE WHICH
         IS A WHOLE NUMBER WITH UP TO TWO DIGITS.  IT PRINTS THE
         PREVIOUS AND PRESENT CONTENTS OF THE NUMBER.  THE
         ARGUMENTS PASSED TO NUMBER ARE THE NAME AND THE NUMBER
         OF DIGITS OF THE NUMBER.
                                                               */
NUMBER:  PROCEDURE(NUM,LEN);
         PUT FILE(SYSPRINT) EDIT(ITEM)(COL(26),F(2));
         PUT FILE(SYSPRINT) EDIT(NUM)(X(7),F(LEN));
         GET FILE(SYSIN) EDIT(NUM)(X(3),F(LEN));
         PUT FILE(SYSPRINT) EDIT(NUM)(COL(85),F(LEN));
         GO TO NEXT;
         END;
         /*

         TIME UPDATES THE CONTENTS OF A DATE WHICH CONSISTS OF
         THE YEAR AND QUARTER.  IT PRINTS THE PREVIOUS AND
         PRESENT CONTENTS OF THE DATE.  THE ARGUMENT PASSED TO
         TIME IS THE NAME OF THE DATE.
                                                               */
TIME:    PROCEDURE(DATE);
         DECLARE 1 DATE,
                   2 YR FIXED BINARY,
                   2 QTR FIXED BINARY;
         PUT FILE(SYSPRINT) EDIT(ITEM,DATE)(COL(26),F(2),X(7),
           F(2),X(2),F(1));
         GET FILE(SYSIN) EDIT(DATE)(X(3),F(2),X(3),F(1));
         PUT FILE(SYSPRINT) EDIT(DATE)(COL(85),F(2),X(2),F(1));
         GO TO NEXT;
         END;
FINISH:  END;
```

APPENDIX D

CEP MANAGEMENT INFORMATION SYSTEM

USERS' MANUAL

TABLE OF CONTENTS

Page

INTRODUCTION. 135
PROGRAMS AVAILABLE IN THE SYSTEM. 136
GENERAL USER INFORMATION. 137
GENERAL REQUEST FORM. 139
ALL 140
BASIC 147
BUILD 151
CHECK 160
FILE. 164
GRADS 167
LOCAL 171
PRINT 174
PURGE 182
REQUEST 186
SS. 189
TEAM. 192
UPDATE. 195

INTRODUCTION

The CEP Management Information System (MIS) consists of programs which manipulate student records in the main file. The result is to print, in a variety of forms, information contained within the file.

The users' manual presents in alphabetical order by program name the basic information needed in order to execute programs in the system. Each program description includes the following items: output description, class size, number of tape drives, DD statements, data setup, data recording or request form if applicable, sample data deck, and sample output.

It is assumed that the user of the system is minimally familiar with both Job Control Language (JCL) and Programming Language/One (PL/1).

PROGRAMS AVAILABLE IN THE SYSTEM

PROGRAM NAME	DESCRIPTION
ALL	Prints all information contained in the main file on those students as requested by faculty members.
BASIC	Prints personal information on students which should be checked and corrected each quarter because it tends to change.
BUILD	Adds new student records to the main file.
CHECK	Prints information about those students who expect to graduate during a specified quarter.
FILE	Prints an ascending list of the student records in the main file by social security number.
GRADS	Deletes records of CEP graduates from the main file.
LOCAL	Prints an alphabetical list of the local addresses of team members by seminar.
PRINT	Prints specific data on those students as requested by faculty members.
PURGE	Deletes records of non-registered students from the main file.
REQUEST	Prints personal information of those students requesting field experience.
SS	Prints an alphabetical list of team members and their social security numbers by seminar.
TEAM	Prints an alphabetical list of team members by seminar.
UPDATE	Updates information in records contained in the main file.

GENERAL USER INFORMATION

The programs in the MIS should be executed using the following JCL cards (see p. 138). All underscored entries are to be supplied by the user.

One or all of the DD statements will be used depending upon the program being executed. Only those DD cards which are specified for a program should be included since each PL/1 file name is associated with the ddname of a DD statement which is then associated with a specific dsname.

All numbers punched on the data cards should be right justified in the columns reserved for that number. For example, many programs in the system require a data card containing in columns 1-2 the number of teams to be printed. If one team is to be printed, this number should be punched _1 in column 2.

All data items not described as being numbers are either character or bit strings. String data should be left justified in the columns reserved for that string. For example, all programs in the system require a data card containing the current date in columns 1-8. If the current date is 1/1/75, this character string should be punched 1/1/75__ on the card in columns 1-6.

Dictionaries for the various codes recorded in the structure PERSONAL are included at the end of the manual.

The general request form (see p. 139) can be used by the team leader to obtain output from the following programs: BASIC, CHECK, FILE, LOCAL, REQUEST, SS, AND TEAM.

JCL CARDS

```
COLUMN   0        1         2         3         4         5         6         7         8
NUMBER   12345678901234567890123456789012345678901234567890123456789012345678901234567890

         //JOBNAME  JOB (____,_____,___,__,____),'PROGRAMMER NAME',CLASS=_
         /*PASSWORD       _,PASSWORD
         /*SETUP          TAPE9,_
         // EXEC PL1CX_
         //PL1.SYSIN DD *
           .
           .
           .
           .
           .    SOURCE PROGRAM
           .
           .
           .
           .
         /*
         //GO.OLD     DD UNIT=TAPE9,VOL=SER=____,DISP=(,KEEP),DCB=(RECFM=FB,
         //              LRECL=3693,BLKSIZE=3693,DEN=3),LABEL=(1,SL),DSN=____
         //GO.NEW     DD UNIT=TAPE9,VOL=SER=____,DISP=(,KEEP),DCB=(RECFM=FB,
         //              LRECL=3693,BLKSIZE=3693,DEN=3),LABEL=(1,SL),DSN=____
         //GO.DELETE DD UNIT=TAPE9,VOL=SER=____,DISP=(,KEEP),DCB=(RECFM=FB,
         //              LRECL=____,BLKSIZE=____,DEN=3),LABEL=(1,SL),DSN=____
         //GO.SYSIN DD *
           .
           .
           .
           .
           .    DATA DECK
           .
           .
           .
           .
         /*
```

NOTE: PARM = 'EXTDICT' should be specified on the execute card for the following programs: BUILD, PRINT, and UPDATE.

GENERAL REQUEST FORM

Team Number: ____________________

Date: ____________________________

Directions: Please check the name of the program which prints the information you currently need. Indicate other information if applicable.

_____BASIC	Prints personal information on students which should be checked and corrected each quarter because it tends to change.
_____FILE	Prints an ascending list of the student records in the main file by social security number.
_____LOCAL	Prints an alphabetical list of the local addresses of team members by seminar.
_____SS	Prints an alphabetical list of team members and their social security numbers by seminar.
_____TEAM	Prints an alphabetical list of team members by seminar.
_____CHECK	Prints information about those students who expect to graduate during a specified quarter.

Graduation date: ________________
Year (last two digits): ____________
Quarter: (Please check the quarter)
Fall _____(1) Spring _____(3)
Winter _____(2) Summer _____(4)

_____REQUEST	Prints personal information on those students requesting field experience.

Date for upcoming field experience: ______________________________

ALL

Output Description	Prints all information contained in the main file on those students as requested by faculty members.
Class	M
Tape Drives	1
DD Statements	OLD

Date Setup

Card	Contents	Column
1	Lowest team number to be printed	1
	Highest team number to be printed	6
2	Date (e.g. 1/13/75)	1-8
3	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47
One card for each request	Faculty code number	1-2
	Request type code number which is a print status code associated with the program named PRINT	6
	Team number	11
	Seminar number	16

There may be more than one faculty request. If there are multiple faculty requests, one request set should follow the other.

If the request type is zero the following cards must follow the request.

Card	Contents	Column
One card for each individual request	Social security number of the student	1-9
After all social security numbers	*	1
Next request		

FACULTY REQUEST
ALL

Request for ____________________ for information on CEP students.
(faculty members name)

Request type (check one)

____Seminar group

Team Number:______________
Seminar:__________________

____Individuals
Please list the social security numbers of those students you wish to include in this request.

__________	__________	__________
__________	__________	__________
__________	__________	__________
__________	__________	__________
__________	__________	__________

SAMPLE DATA DECK FOR ALL

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890

          3    3
          1/13/75
           1
          3    1    1     8
           8   0    3    1
          965573617
          *
```

```
1/13/73
AL
OUTPUT DESCRIPTION: LISTING OF ALL DATA ON STUDENTS AS REQUESTED BY FACULTY MEMBERS
REQUEST FOR KINZER


[illegible]_NUMBER_AND_NAME           ITEM           CONTENTS

966972617
     GRANT.MORRI                HOME ADDR.     1976 HIGH STREET #2
                                               SUNLAND. CA 91040

                                HOME PHONE     7145339117
                                LOCAL ADDR.    77 SUNSET DRIVE
                                               SUNRISE. FL 33313

                                LOCAL PHONE    3765935

                                SEX            FEMALE

                                BIRTHDAY        2/20/53

                                MARITAL STATE MARRIED

                                DEP. CHILDREN  2

                                AGE RANGE      YOUNGEST =  1   OLDEST =  3

                                WORK EXP.      BABYSITTER                  CAMP COUNSELOR              RECREATION CENTER
                                                   NURSERY SCHOOL

                                TRAVEL EXP.    MEXICO                      U.S.A.- ALL OVER          CARIBBEAN

                                EMPLOYMENT     EMPLOYER = CITY OF FROSTPROOF     HOURS WORKED PER WEEK = 20

                                COLLEGES

                                              .INSTITUTION = MIAMI DADE JUNIOR COLLEGE        FROM = 72 SU    TO = 73 W

                                               INSTITUTION = UNIVERSITY OF FLORIDA            FROM = 74 1     TO = CUR

                                GPA            2.6 - 3.0

                                MYERS BRIGGS   INTJ

                                CEP_ENTRANCE   YEAR =        74   QUARTER = FALL     HOURS COMPLETED =  91.0

                                CONCENTRATION ELEMENTARY

                                TEAM           3

                                SEMINAR        KINZER

                                GRAD. DATE     YEAR = 76   QUARTER = SPRING

                                REQ. COURSES   GRADE = * INDICATES COURSE REQUIREMENT HAS BEEN MET
                                               EH491                 GRADE = *
                                               MSC260                GRADE = *
                                               MSC390                GRADE =
                                               MS310/311             GRADE = *
                                               PHR361                GRADE = *
                                               PHR373                GRADE =
                                               SCH201                GRADE = *
                                               MEDIA EQUIPMENT       GRADE =
                                               MEDIA PRODUCTIONS     GRADE =

                                REGISTRATION   YR = 74   QTR = 1   EDE311     HRS =  8  GRADE = S
                                                                   EDL405     HRS =  2  GRADE = S
                                                                   MSC390     HRS =  2  GRADE = B
                                                                   PL262      HRS =  2  GRADE = A
                                               YR = 75   QTR = 2   EDE405     HRS =  2  GRADE =
                                                                   EDE322     HRS =  8  GRADE =

                                L.A.           AREA = ART                      NO =  1   10/ 8/74  SMITH
                                                   ORIENTATION                                          INDIVIDUAL
```

```
1/13/75
ALL
OUTPUT DESCRIPTION: LISTING OF ALL DATA ON STUDENTS AS REQUESTED BY FACULTY MEMBERS
REQUEST FOR KINZER


S.S. NUMBER AND NAME           ITEM            CONTENTS
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = ART                          NO = 4   10/18/74  NEWCOMB
                                                  ART CURRICULUM PLANNING                                   INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =     ACCURATE
                                                                 BRIEF
                                                                 COMPETENT
                                                                 DESCRIPTIVE
                                                                 GOOD CLASSROOM APPLICATIONS
                                                                 SYSTEMATIC
                                             AREA = CURRICULUM                   NO = I   11/ 5/74  SHEA
                                                  ORIENTATION                                               INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = HUMAN GROWTH AND DEV.        NO = 1   10/ 5/74  JESTER
                                                  ACTIVITY 2                                                INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = LANGUAGE ARTS                NO = 1   11/28/74  FEY
                                                  ORIENTATION                                               INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = MATH                         NO = 1   10/26/74  OLSON
                                                  ORIENTATION                                               INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = MUSIC                        NO = 1    9/28/74  PENLAND
                                                  ORIENTATION                                               INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = READING                      NO = 1   10/15/74  FEY
                                                  ORIENTATION                                               INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = READING                      NO = 2   10/16/74  LEVINE
                                                  TERMINOLOGY AND WORD ATTACK SKILLS                        INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =     ACCURATE
                                             AREA = SCIENCE                      NO = 1   11/ 3/74  DETURE
                                                  LABORATORY 1                                              INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =     DESCRIPTIVE
                                                                 CAREFUL IDEAS
                                             AREA = SCIENCE                      NO = 2   10/17/74  KRALL
                                                  LABORATORY 2                                              INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =     SATISFACTORY
                                             AREA = SOCIAL FOUNDATIONS           NO = 1   11/ 4/74  CURRAN
                                                  ORIENTATION                                               INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =     WORK COMPETENT IF ATTENTIVE
                                             AREA = SOCIAL FOUNDATIONS           NO = 2   11/22/74  CURRAN
                                                  KEEPING UP WITH RESEARCH                                  INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =
                                             AREA = SOCIAL STUDIES               NO = 1    9/28/74  HILLIARD
                                                  ORIENTATION                                               INDIVIDUAL
                                                  REGULAR
                                                  COMMENTS =     INTERESTING
                                                                 STIMULATING
                               FIELD         YR = 74   QTR = 1   HAWTHORNE      BLUE
                                                  GRADE = THIRD                     OPEN
                                                  ROLE =         INDIVIDUAL INSTRUCTION
                                                                 SMALL GROUP INSTRUCTION
                                                  STRENGTHS =    AWARE OF INDIVIDUAL NEEDS
                                                                 AWARE OF INDIVIDUAL PROBLEMS
                                                                 GOOD WITH CHILDREN
```

```
1/,3/75
ALL
OUTPUT DESCRIPTION: LISTING OF ALL DATA ON STUDENTS AS REQUESTED BY FACULTY MEMBERS
REQUEST FOR KINZER


CASE NUMBER AND NAME       ITEM          CONTENTS

                                            NEEDS =          SENSITIVE TO STUDENTS FEELINGS
                                                             TO PREPARE UNITS
                                                             SMALL GROUP EXPERIENCE
                                                             ART EXPERIENCE
                                                             P.E. EXPERIENCE

                           COMMUNITY     YR = 75   QTR = 2   GATOR NUSERY               AGENCY REP =    DOE
                                         ASSISTANT
                                            ROLE =           GENERAL NURSERY ACTIVITIES
                                                             ORGANIZED GAMES
                                                             PLAYED WITH CHILDREN
                                            COMMENTS =       INITIATIVE
                                                             WORKS WELL WITH CHILDREN
```

BASIC

Output Description	Prints personal information on students which should be checked and corrected each quarter because it tends to change.
Class	S
Tape Drives	1
DD Statements	Old

Data Setup

Card	Contents	Column
1	Lowest team number to be printed Highest team number to be printed	1 6
2	Date (e.g. 1/1/75)	1-8
3	Return date (e.g. Jan. 13, 1975)	1-15
4	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47

SAMPLE DATA DECK FOR BASIC

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890


          3    3
          1/13/75
          JANUARY 18,1975
           1
          3    1    1     8
```

```
1/13/75
BASIC
OUTPUT DESCRIPTION: LISTING OF PERSONAL INFORMATION ON STUDENTS WHICH IS TO CHECKED AND CORRECTED BY THE STUDENTS AT THE BEGINNING
 OF EACH QUARTER.
RETURN DATE: JANUARY 18,1975
SEMINAR LEADER: KINZER
```

```
DIRECTIONS: PLEASE CORRECT ANY INFORMATION BELOW WHICH IS INCORRECT.
    RETURN THIS FORM TO YOUR SEMINAR LEADER NO LATER THAN JANUARY 16,1976.

SOCIAL SECURITY NUMBER(1) :             966973617

NAME(2) :
        LAST(3) :                       BEANE
        FIRST(4) :                      MERRI
        MIDDLE(5) :                     MEAN

PERMANENT ADDRESS(6) :
        STREET(7) :                     1976 HIGH STREET #2
        CITY(8) :                       SUNLAND
        STATE(9) :                      CA
        ZIP CODE(10) :                  91040

PERMANENT PHONE NUMBER(11) :            7146539117

LOCAL ADDRESS(12) :
        STREET(13) :                    77 SUNSET DRIVE
        CITY(14) :                      SUNRISE
        STATE(15) :                     FL
        ZIP CODE(16) :                  33312

LOCAL PHONE NUMBER(17) :                3765935

MARITAL STATUS(23) :                    PLEASE CHECK YOUR CURRENT MARITAL STATUS.
                                             SINGLE                        ( 1)
                                             MARRIED              .      X ( 2)
                                             DIVORCE                       ( 3)
                                             WIDOWED                       ( 4)

NUMBER OF DEPENDENT CHILDREN(24) :       2

AGE RANGE OF DEPENDENT CHILDREN(26) :
        YOUNGEST CHILD :                          1
   .    OLDEST CHILD :                            3

WORK EXPERIENCE(26) :                   PLEASE CHECK THE TYPES OF WORK EXPERIENCES YOU HAVE HAD.
                                             BABYSITTER                  X ( 1)       RECREATION CENTER            X (10)
                                             CAMP COUNSELOR              X ( 2)       SALESPERSON                    (11)
                                             CLERK                         ( 3)       SECRETARY                      (12)
                                             CLUB WORK                     ( 4)       STUDENT ASSISTANT              (13)
                                             DAY CARE CENTER               ( 5)       SUBSTITUTE                     (14)
                                             LABORATORY TECHNICIAN         ( 6)       TEACHERS AID                   (15)
                                    .        LITTLE THEATRE                ( 7)       TOUR GUIDE                     (16)
                                             NURSERY SCHOOL              X ( 8)       TUTOR                          (17)
                                             RECEPTIONIST                  ( 9)       OTHER WORK CLASSIFICATION      (18)

TRAVEL EXPERIENCE(28) :                 PLEASE CHECK THE TYPES OF TRAVEL EXPERIENCES YOU HAVE HAD.
                                             AFRICA                        ( 1)       U.S.A.- ALL OVER             X (11)
                                             ASIA                          ( 2)       FAR WESTERN STATES             (12)
                                             BERMUDA AND BAHAMAS           ( 3)       MIDDLE ATLANTIC STATES         (13)
                                             CANADA                        ( 4)       MIDWESTERN STATES              (14)
                                             CARIBBEAN                   X ( 5)       NEW ENGLAND STATES             (15)
                                             CENTRAL AMERICA               ( 6)       PLAINS STATES                  (16)
                                             EUROPE                        ( 7)       ROCKY MOUNTAIN STATES          (17)
                                             MEXICO                      X ( 8)       SOUTHEASTERN STATES            (18)
                                             MIDDLE EAST                   ( 9)       SOUTHWESTERN STATES            (19)
                                             PACIFIC ISLANDS               (10)                                      (20)

CURRENT EMPLOYMENT(30) :
        EMPLOYER(31) :                  CITY OF FROSTPROOF
        HOURS WORKED PER WEEK(32) :     20

MYERS BRIGGS TYPE(39) :                 INTJ

SPECIALIZATION(44) :                    ELEMENTARY

TEAM(45) :                               3

SEMINAR(46) :                           KINZER

EXPECTED DATE OF GRADUATION(47) :
        YEAR(48) :                      76
        QUARTER(49) :                   SPRING
```

BUILD

Output Description	Adds new student records to the main file.
Class	S
Tape Drives	2
DD Statements	OLD NEW

Data Setup

Card	Contents	Column
1	Type code number	1
2	Date (e.g. 1/1/75)	1-8
One card set for each student		
1	Social security number	1-9
2	Number two	1-2
	Name:	
	Last	6-19
	First	21-30
	Middle	36-49
3	Number six	1-2
	Permanent address:	
	Street	6-25
	City	31-50
	State	56-57
	Zip code	61-65
4	Number eleven	1-2
	Permanent phone number	6-15
5	Number twelve	1-2
	Local address:	
	Street	6-25
	City	31-50
	State	56-57
	Zip code	61-65
6	Number seventeen	1-2
	Local phone number	6-12
7	Number eighteen	1-2
	Sex code	6

Card	Contents	Column
8	Number nineteen	1-2
	Birthdate:	
	Month number	6-7
	Day	11-12
	Year (last two digits)	16-17
9	Number twenty-three	1-2
	Marital status code number	6
10	Number twenty-four	1-2
	Number of dependent children	6
11	Number twenty-five	1-2
	Age range of the dependent children:	
	Youngest	6-7
	Oldest	11-12
12	Number twenty-six	1-2
	First work experience code number	6-7
	Second work experience code number	11-12
	Third work experience code number	16-17
	Fourth work experience code number	21-22
	Fifth work experience code number	26-27
	Sixth work experience code number	31-32
13	Number twenty-eight	1-2
	First travel experience code number	6-7
	Second travel experience code number	11-12
	Third travel experience code number	16-17
	Fourth travel experience code number	21-22
	Fifth travel experience code number	26-27
	Sixth travel experience code number	31-32
14	Number thirty	1-2
	Current employment:	
	Employer	6-25
	Hours worked per week	31-32
15	Number thirty-three (altered to input information on all colleges attended)	1-2
	Number of colleges attended	6
One card for each college attended (include the first college on card 14)	College name	11-40
	Attendance dates:	
	Beginning (i.e. 74 F)	46-51
	Ending	56-61
16+	Number thirty-eight	1-2
	GPA code number	6

Card	Contents	Column
17+	Number thirty-nine	1-2
	Myers Briggs type code number	6-7
18+	Number forty	1-2
	CEP entrance date:	
	Year (last two digits)	6-7
	Quarter code number	11
	Hours entering program	16-19
19+	Number forty-four	1-2
	Specialization code number	6
20+	Number forty-five	1-2
	Team number	6
21+	Number forty-six	1-2
	Seminar number	6
22+	Number forty-seven	1-2
	Expected graduation date:	
	Year (last two digits)	6-7
	Quarter code number	11
23+	Number fifty (altered to input all required course information)	1-2
	Grade for EH 491	6
	Grade for MSC 260	11
	Grade for MSC 390	16
	Grade for MS 310 or MS 311	21
	Grade for PHR 361	26
	Grade for PHR 373	31
	Grade for SCH 201	36
	Grade for Media A	41
	Grade for Media B	46
24+	*(trailer label)	1

After executing build sort the records in the file in ascending order by social security number. The system supplied sort merge routine may be used.

INFORMATION ON NEW STUDENTS

Directions: Please complete the information below.

Social security number (1): ____________________

Name (2):
    Last: ____________________
    First: ____________________
    Middle: ____________________

Permanent address (6):
    Street: ____________________
    City: ____________________
    State: ____________________
    Zip code: ____________________

Permanent phone number (11): ____________________

Local address (12):
    Street: ____________________
    City: ____________________
    State: ____________________
    Zip code: ____________________

Local phone number (17): ____________________

Sex (18): Please check your sex.
    Female _____(F)
    Male _____(M)

Birthdate (19):
    Month number: ____________________
    Day: ____________________
    Year (last two digits): ____________________

Marital status (23): Please check your current marital status:
    Single _____(1)
    Married _____(2)
    Divorced _____(3)
    Widowed _____(4)

Number of dependent children (24): ____________________

Age range of dependent children (25):
    Youngest: ____________________
    Oldest: ____________________

Work experience (26):

Please check the types of work experience you have had.

Babysitter _____(1)
Camp counselor _____(2)
Clerk _____(3)
Club work _____(4)
Day care center _____(5)
Laboratory technician _____(6)
Little theatre _____(7)
Nursery school _____(8)
Receptionist _____(9)
Recreation center _____(10)
Salesperson _____(11)
Secretary _____(12)
Student assistant _____(13)
Substitute _____(14)
Teachers aid _____(15)
Tour guide _____(16)
Tutor _____(17)
Other work classification _____(18)

Travel experience (28):

Please check the types of travel experiences you have had.

Africa _____(1)
Asia _____(2)
Bermuda and Bahamas _____(3)
Canada _____(4)
Caribbean _____(5)
Central America _____(6)
Europe _____(7)
Mexico _____(8)
Middle East _____(9)
Pacific islands _____(10)
U.S.A. - all over _____(11)
Far western states _____(12)
Middle Atlantic states _____(13)
Midwestern states _____(14)
New England states _____(15)
Plains states _____(16)
Rocky Mountain states _____(17)
Southeastern states _____(18)
Southwestern states _____(19)
World - all over _____(20)

Current work experience (30):

Employer: ______________________

Hours worked per week: ___________

Colleges attended (33):

Please list the information in chronological order.

College	Beginning attendance date	Ending attendance date

Current grade point average (38): Please check your current grade point average.

1.5 or less _____(1)
1.6 - 2.0 _____(2)
2.1 - 2.5 _____(3)
2.6 - 3.0 _____(4)
3.1 - 3.5 _____(5)
3.6 - 4.0 _____(6)

Myers Briggs Type (39): Please check your Myers Briggs Type.

ISTJ _____(1)
ISTP _____(2)
ESTP _____(3)
ESTJ _____(4)
ISFJ _____(5)
ISFP _____(6)
ESFP _____(7)
ESFJ _____(8)
INFJ _____(9)
INFP _____(10)
ENFP _____(11)
ENFJ _____(12)
INFJ _____(13)
INTP _____(14)
ENTP _____(15)
ENTJ _____(16)

CEP entrance date (40):

Year (last two digits): _______________

Quarter: Please check the quarter you entered CEP.

Fall _____(1)
Winter _____(2)
Spring _____(3)
Summer _____(4)

Specialization (44): Please check your specialization in CEP.

Elementary _____(1)
Early childhood _____(2)

Team (45): ____________________

Seminar (46): ____________________

Expected date of graduation (47):
Year (last two digits): ________
Quarter:

Please check the quarter you expect to graduate.

Fall _____(1)
Winter _____(2)
Spring _____(3)
Summer _____(4)

Required courses (50):

Please check the required courses you have completed here or at another institution. List the grade you received if you remember it.

EH 491 _____(1)
MSC 260 _____(2)
MSC 390 _____(3)
MS 310 or MS 311 _____(4)
PHR 361 _____(5)
PHR 373 _____(6)
SCH 201 _____(7)
Media A _____(8)
Media B _____(9)

SAMPLE DATA DECK FOR BUILD

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890

          1
          1/13/75
          192837645
           2   MEAN          MARY
           6   9 SHADY LANE             GOULDS                   FL   33170
          11     9043297926
          12   32 SNOW HILL             FROSTPROOF               FL   33843
          17   3788341
          18   F
          19    6   17   53
          23   1
          24
          25
          26    1    3   11
          28    7   11
          30   STATE OF FLORIDA          5
          33   1    VALENCIA COMMUNITY COLLEGE         71 F      73 W
          38   3
          39   11
          40   74   1
          44   2
          45   2
          46   2
          47   75   3
          50   *    *              *         *
          *
```

```
1/13/75
BUILD
OUTPUT DESCRIPTION: A LISTING BY SOCIAL SECURITY NUMBER OF ALL DATA ENTERED DURING THE ADDITION OF NEW RECORDS


SOCIAL SECURITY NO       ITEM      PRESENT CONTENTS

    192037645               2      MEAN              MARY
                            6      9 SHADY LANE             GOULDS                FL  33170
                           11       904329792
                           12      32 SNOW HILL             FROSTPROOF            FL  33843
                           17      3788341
                           18      F
                           19       6  17  53
                           23      1
                           24      0
                           25       0   0
                           26       1   3  11   0   0   0
                           28       7  11   0   0   0   0
                           30      STATE OF FLORIDA         5
                           33      VALENCIA COMMUNITY COLLEGE      71 F    73 W
                           35      3
                           39      11
                           40      74   4    0.0
                           44      2
                           45      2
                           46      2
                           47      75   3
                           50      *  *       *     *
```

CHECK

Output Description	Prints information about those students who expect to graduate during a specified quarter.
Class	S
Tape Drives	1
DD Statements	OLD

Data Setup

Card	Contents	Column
1	Lowest team number to be printed Highest team number to be printed	1 6
2	Date (e.g. 1/13/75)	1-8
3	Graduation date:	
	Year (last two digits)	1-2
	Quarter code number	6
4	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47

SAMPLE DATA DECK FOR CHECK

```
COLUMN     0        1         2         3         4         5         6         7         8
NUMBER     12345678901234567890123456789012345678901234567890123456789012345678901234567890


           3    3
           1/13/75
           75   3
            1
           3    1    1     8
```

```
1/15/75
CHECK
GRADUATION LIST:  YR = 75  QTR = 3
OUTPUT DESCRIPTION: MISSING LEARNING ACTIVITIES BY AREA AND TOTAL NUMBER OF MISSING ACTIVITIES
TEAM = 3  SEMINAR = KINZER

                          AREA__1 AREA__2 AREA__3 AREA__4 AREA__5 AREA__6 AREA__7 AREA__8 AREA__9 AREA_10 AREA_11 TOTAL

EADELE.BROOKE                1       1       8       1       1       1       1       1       1       1       1
                             2       2       9       2       2       2       3       2       2       2       2
                             3       3      10       3       3       3       4       3       4       3       3
                             4       4      11       4       4       4       5       4       5       4       0
                             5       5      12       5       5       5       6       6       0       5       0    80

DOSS.BRENDA                  1       5       1       2       2       1       2       1       1       2       1
                             2       6       2       3       3       2       3       2       4       3       2
                             3       8       3       4       4       3       4       3       5       4       3
                             4      10       4       5       5       5       5       4       0       5       0
                             5      11       5       6       6       6       6       5       0       6       0    77

FANG.FAYE                    2       1       1       1       4       1       1       1       1       2       1
                             3       2       2       2       5       2       2       2       2       3       2
                             4       3       3       3       6       5       4       3       3       4       3
                             5       4       4       4       7       6       5       4       4       5       0
                             6       5       5       5       8       0       6       5       5       6       0    86

SHOWERS.APRIL                1       1       9       1       4       1       1       1       1       3       1
                             5       2      10       3       5       2       3       2       4       4       2
                             6       4      12       4       6       3       4       3       5       5       3
                             0       5       0       5       7       4       5       6       0       6       0
                             0       6       0       6       8       5       7       7       0       7       0    69

SWINDLER.JACKIE              1       1       9       1       2       1       1       1       4       2       1
                             2       2      10       2       3       2       2       2       5       3       2
                             3       3      11       3       4       3       3       3       0       4       3
                             4       4      12       4       5       4       4       4       0       5       0
                             5       5       0       5       6       5       5       5       0       6       0    85
```

```
1/13/75
CHECK
GRADUATION LIST:  YR = 75  QTR = 3
OUTPUT DESCRIPTION: ALPHABETICAL LIST OF EXPECTED GRADUATES IN A TEAM
   BY SEMINAR.  STUDENTS WITH MORE THAN 5 LEARNING ACTIVITIES TO COMPLETE ARE MARKED BY AN *
TEAM  3



     KINZER__________

    *BABBLE.BROCKE
    *BOSS.BRENDA
    *FANC.FAYE
    *SHOWENS.APRIL
    *SWINDLER.JACKIE
```

FILE

Output Description	Prints an ascending list of the student records in the main file by social security number.
Class	S
Tape Drives	1
DD Statements	OLD

Data Setup

Card	Contents	Column
1	Lowest team number to be printed	1
	Highest team number to be printed	6
2	Date (e.g. 1/13/75)	1-8
3	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47

SAMPLE DATA DECK FOR FILE

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890


          3    3
          1/13/75
           1
          3    1    1     8
```

```
1/13/75
FILE
OUTPUT DESCRIPTION: ASCENDING LIST OF THE STUDENT RECORDS IN THE MAIN FILE BY SOCIAL SECURITY NUMBER

SOCIAL_SECURITY_NO     NAME                          TEAM     SEMINAR
133975253              SHOWERS.APRIL                 3        KINZER
113428170              DARLING,DAVID                 3        KINZER
121717683              JOLLEY.HOLLY                  3        KINZER
122909937              LORD.LULU                     3        KINZER
155533426              FLORA,ROSE                    3        KINZER
196450930              BEACH,SANDI                   3        KINZER
205361040              TOPPER,TERRI                  3        KINZER
229211687              PEPPERCORN.CORNELIUS          3        KINZER
232090255              HOOD.RUBIN                    3        KINZER
245002403              WACK.WILLA                    3        KINZER
310501090              SWINDLER,JACKIE               3        KINZER
340727685              ZEPPER.ZILDA                  3        KINZER
340935169              SMART.SALLY                   3        KINZER
346735487              LIES.LILY                     3        KINZER
350303360              KEAN.ELIZABETH                3        KINZER
353034261              LUST,LORNA                    3        KINZER
375720456              HORNE.HILDA                   3        KINZER
383113116              FANG.FAYE                     3        KINZER
542927494              ADAMS.EVE                     3        KINZER
635733213              GLORY.VICTORIA                3        KINZER
644323665              DRIGGERS.DRUSILLA             3        KINZER
645947429              BABBLE,BROOKE                 3        KINZER
660657471              GRUMP.GRENDA                  3        KINZER
737404575              BOSS,BRENDA                   3        KINZER
765201358              KLINKER.CLARIBEL              3        KINZER
801573614              FERN.FRANCES                  3        KINZER
809550110              GRUBB.OLGA                    3        KINZER
852697760              BENZ.MERCEDES                 3        KINZER
966573517              LEANE.MERRI                   3        KINZER
```

GRADS

Output Description	Deletes records of CEP graduates from the main file.
Class	S
Tape Drives	3
DD Statements	OLD NEW DELETE LRECL=94, BLKSIZE=6392

Data Setup

Card	Contents	Column
1	Date (e.g. 1/13/75)	1-8
2	Graduation date:	
	Year (last two digits)	1-2
	Quarter code number	6
One card for each graduate	Social security number of the graduate	1-9

Graduates _must_ be listed in ascending order by social security number.

CEP GRADUATES

Team Number: ______________________________

Date: ____________________________________

Directions: Please list in ascending order by social security number the name and associated social security number of all students who have graduated this quarter.

Name	Social security number

SAMPLE DATA DECK FOR GRADS

```
COLUMN  0         1         2         3         4         5         6         7         8
NUMBER  12345678901234567890123456789012345678901234567890123456789012345678901234567890

        1/13/75
        70   3
        100975253
        111111111
        159533426
        229511687
```

```
1/14/75
GRADS
OUTPUT DESCRIPTION: ASCENDING LIST OF GRADUATES WHOSE RECORDS WERE PURGED
     FROM THE MAIN FILE AND DESTROYED UNLESS OTHERWISE INDICATED


ERROR MESSAGE      S.S.NUMBER      NAME                        ADDRESS
                   105975283      SHOWERS,APRIL               24 RAINDROP DRIVE     BALM              FL  33503
MISSING            111111111
INCORRECT DATE     139533426
                   229511467      PEPPERCORN,CORNELIUS        48 SPICEWOOD DRIVE    GREENACRES        FL  33406
```

LOCAL

Output Description	Prints an alphabetical list of the local addresses of team members by seminar.
Class	S
Tape Drives	1
DD Statements	OLD

Data Setup

Card	Contents	Column
1	Lowest team number to be printed	1
	Highest team number to be printed	6
2	Date (e.g. 1/13/75)	1-8
3	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47

SAMPLE DATA DECK FOR LOCAL

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890

          3    3
          1/13/75
           1
          3    1    1     8
```

```
12/3/75
LOCAL
OUTPUT DESCRIPTION: ALPHABETICAL LIST OF TEAM MEMBERS ADDRESSES BY SEMINAR
TEAM   3

KINZIE________

ADAMS.EVE
39 FIRST AVENUE
FROSTPROOF
3929459

BABBLE.BROOKE
64 PARADISE TRAIL
FROSTPROOF
3736023

BEACH.SANDI
31 BAY BOULEVARD
FROSTPROOF
3730938

BEANE.MERRI
77 SUNSET DRIVE
SUNRISE
3765935

BENZ.MERCEDES
11 KING STREET
FROSTPROOF
3774028

BOSS.BRENDA
1 MASTERS DRIVE
FROSTPROOF
3929784

DARLING.DAVID
95 MAIN STREET
FROSTPROOF
3763998

DRIGGLES.DRUSILLA
100 WINNERS CIRCLE
FROSTPROOF
3777121

FANG.FAYE
49 DRUID PLACE
FROSTPROOF
3739675

FERN.FRANCES
61 FLORIST AVENUE
FROSTPROOF
3737662

FLORA.ROSE
65 GARDEN TRAIL
FROSTPROOF
3927634

GLORY.VICTORIA
75 VICTORY LANE
FROSTPROOF
3776111

GRUBB.OLGA
3 LEROY AVENUE
FROSTPROOF
3737133

GRUMP.GRENDA
55 FOWLER AVENUE
FROSTPROOF
3775494

HOOD.ROBIN
125 FRIAR TUCK AVE
FROSTPROOF
3778903

HORNE.HILDA
12 HIGH STREET
FROSTPROOF


JOLLEY.HOLLY
60 HOLIDAY ROAD
FROSTPROOF
7550542

KEAN.ELIZABETH
55 COOLEY DRIVE
FROSTPROOF
4682768

KLINKER.CLARIBEL
13 BIRD DRIVE
FROSTPROOF
3738227

LIES.LILY
72 LAW LANE
FROSTPROOF
3760756

LOBO.LULU
44 PAYNE PLACE
FROSTPROOF
3735634

LOST.LORNA
2 WINDING ROAD
FROSTPROOF
4663709

PEPPERCORN.CORNELIUS
99 VINE STREET
FROSTPROOF


SHOWERS,APRIL
41 MAY STREET
FROSTPROOF
3731673

SMART.SALLY
99 BRIGHTSIDE PLACE
FROSTPROOF
3772945

SWINDLER.JACKIE
17 LAKE DRIVE
FROSTPROOF
3736023

TOPPER.TERRI
53 HILLTOP ROAD
FROSTPROOF
3769297

WACK.WILLA
83 CHERRY DRIVE
FROSTPROOF


ZEPPER,ZILDA
30 BEDROCK ROAD
FROSTPROOF
7627119
```

PRINT

Output Description	Prints specific data on those students as requested by faculty members.
Class	M
Tape Drives	1
DD Statements	OLD

Data Setup

Card	Contents	Column
1	Date (e.g. 1/13/75)	1-8
Three cards for each request		
1	Faculty code number	1-2
	Request type code number	6
	Team number	11
	Seminar number	16
2	Myers Briggs status code number	1
	Required courses:	
	Status code number	16
	Code number of the required course to be printed (needed for status code 1)	21-22
	Registration:	
	Status code number	31
	Number of the quarter in the program to be printed (needed for status code 1)	36-37
	Date of the registration to be printed (needed for status code 3)	
	Year (last two digits)	36-37
	Quarter code number	41
	Learning activities:	
	Status code number	46
	Area number to be printed (needed for status codes 1 and 2)	51-52
	Activity number to be printed (needed for status code 1)	56-57
	Learning activity summary:	
	Status code number	61
	Area number to be printed (needed for status code 1) .	66-67

Card	Contents	Column
3	Field experiences:	
	Status code number	1
	Number of the field experience to be printed (needed for status code 1)	6-7
	Field experience summary:	
	Status code number	16
	Number of the field experience summary to be printed (needed for status code 1)	21-22
	Community experience status code number	31

There may be more than one faculty request. If there are multiple faculty requests, one request set should follow the other.

If the request type is zero the following cards must follow the request.

Card	Contents	Column
One card for each individual request	Social security number of the student	1-9
After all social security numbers	*	1
Next request		

PRINT STATUS CODES

Code Name	Code Number	Effect
Request type	0	Individual
	1	Group
Myers Briggs	0	Suppress printing
	1	Print
Required courses	0	Suppress printing
	1	Print one required course
	2	Print all required courses which are nonblank
Registration	0	Suppress printing
	1	Print one registration by quarter in the program
	2	Print all registrations which are nonblank
	3	Print one registration by date
Learning activities	0	Suppress printing
	1	Print one learning activity
	2	Print learning activities in one area
	3	Print all nonblank learning activities
Learning activity summary	0	Suppress printing
	1	Print one area summary
	2	Print all area summaries
Field experience	0	Suppress printing
	1	Print one field experience
	2	Print all nonblank field experiences
Field experience summary	0	Suppress printing
	1	Print one field experience summary
	2	Print all nonblank field experience summaries
Community experience	0	Suppress printing
	1	Print

FACULTY REQUEST
PRINT

Request by ____________________ for information on CEP students.
(faculty members name)

For each category below please indicate the information which you want by checking the appropriate response. If additional information is needed for a request (e.g. area name) please fill in the blank below the response with the necessary information.

Request type (check one):
_____Seminar group
Team number:______________
Seminar:__________________
_____Individuals
Please list the social security numbers of those students you wish to include in this request.

______________	______________	______________
______________	______________	______________
______________	______________	______________
______________	______________	______________
______________	______________	______________

1. Myers Briggs Type Indicator (check one)
   - Do not print _____(0)
   - Print _____(1)

2. Required courses (check one)
   - Do not print _____(0)
   - Print a specific required course _____(1)
     - Course name______________
   - Print all required courses _____(2)

3. Registration information (check one)
   - Do not print _____(0)
   - Print a specific registration by quarter in the program (i.e. first, second, . . .) _____(1)
     - Quarter in the program______________
   - Print all quarters registered _____(2)
   - Print a specific registration by date _____(3)
     - Year ______________
     - Quarter # ______________

4. Learning activities (check one)
   - Do not print _____(0)
   - Print a specific learning activity _____(1)
     - Area # ______________
     - Activity # ______________
   - Print all learning activities completed in a specific area _____(2)
     - Area # ______________
   - Print all learning activities completed _____(3)

5. Learning activity summary (check one)
    Do not print ______(0)
    Print a specific area summary. ______(1)
        Area # ____________
    Print all area summaries ______(2)

6. Field experience (check one)
    Do not print ______(0)
    Print a specific field experience ______(1)
        Field experience #____________
    Print all field experiences completed ______(2)

7. Field experience summary (check one)
    Do not print ______(0)
    Print a specific field experience summary ______(1)
        Field experience # ____________
    Print all field experience summaries ______(2)

8. Community experience (check one)
    Do not print ______(0)
    Print ______(1)

SAMPLE DATA DECK FOR PRINT

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890

          1/13/75
           8   0    3    1
          1              2              2              3              2
          2              2              1
          966973617
          *
```

```
1/13/75
PRINT
OUTPUT DESCRIPTION: LISTING OF SPECIFIC DATA ON CERTAIN STUDENTS AS REQUESTED BY FACULTY MEMBERS
REQUEST FOR KINZER


STU. NUMBER AND NAME        ITEM            CONTENTS

366972617
     BEANE,MEHRI            MYERS BRIGGS    INTJ

                            REQ. COURSES    GRADE = * INDICATES COURSE REQUIREMENT HAS BEEN MET
                                            EM491                   GRADE = *
                                            MSC260                  GRADE = *
                                            MSC390                  GRADE =
                                            MS310/311               GRADE = *
                                            PHR361                  GRADE = *
                                            PFR373                  GRADE =
                                            SCH201                  GRADE = *
                                            MEDIA EQUIPMENT         GRADE =
                                            MEDIA PRODUCTIONS       GRADE =

                            REGISTRATION    YR = 74    QTR = 1    EDL311    HRS =  8  GRADE = S
                                                                  EDE405    HRS =  2  GRADE = S
                                                                  MSC390    HRS =  2  GRADE = B
                                                                  PE202     HRS =  2  GRADE = A
                                            YR = 75    QTR = 2    EDE455    HRS =  2  GRADE =
                                                                  EDL322    HRS =  8  GRADE =

                            L.A.            AREA = ART                      NO =  1   10/ 8/74  SMITH
                                                 ORIENTATION                                          INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =
                                            AREA = ART                      NO =  4   10/16/74  NEWCOMB
                                                 ART CURRICULUM PLANNING                              INDIVIDUAL
                                                 REGULAR*
                                                 COMMENTS =     ACCURATE
                                                                BRIEF
                                                                COMPETENT
                                                                DESCRIPTIVE
                                                                GOOD CLASSROOM APPLICATIONS
                                                                SYSTEMATIC
                                            AREA = CURRICULUM               NO =  1   11/ 5/74  SHEA
                                                 ORIENTATION                                          INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =
                                            AREA = HUMAN GROWTH AND DEV.    NO =  1   10/ 5/74  JESTER
                                                 ACTIVITY 2                                           INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =
                                            AREA = LANGUAGE ARTS            NO =  1   11/28/74  FEY
                                                 ORIENTATION                                          INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =
                                            AREA = MATH                     NO =  1   10/26/74  OLSON
                                                 ORIENTATION                                          INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =
                                            AREA = MUSIC                    NO =  1    9/28/74  PENLAND
                                                 ORIENTATION                                          INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =
                                            AREA = READING                  NO =  1   10/15/74  FEY
                                                 ORIENTATION                                          INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =
                                            AREA = READING                  NO =  2   10/16/74  LEVINE
                                                 TERMINOLOGY AND WORD ATTACK SKILLS                   INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =     ACCURATE
                                            AREA = SCIENCE                  NO =  1   11/ 3/74  DETURE
                                                 LABORATORY I                                         INDIVIDUAL
                                                 REGULAR
                                                 COMMENTS =     DESCRIPTIVE
```

```
1/13/75
PRINT
OUTPUT DESCRIPTION: LISTING OF SPECIFIC DATA ON CERTAIN STUDENTS AS REQUESTED BY FACULTY MEMBERS
REQUEST FOR KINZER


S.S. NUMBER AND NAME         ITEM          CONTENTS

                                                       CAREFUL IDEAS
                                           AREA = SCIENCE                  NO =  2   10/17/74  KRALL
                                                LABORATORY 2                                        INDIVIDUAL
                                                REGULAR
                                                COMMENTS =     SATISFACTORY
                                           AREA = SOCIAL FOUNDATIONS       NO =  1   11/ 4/74  CURRAN
                                                ORIENTATION                                         INDIVIDUAL
                                                REGULAR
                                                COMMENTS =     WORK COMPETENT IF ATTENTIVE
                                           AREA = SOCIAL FOUNDATIONS       NO =  2   11/22/74  CURRAN
                                                KEEPING UP WITH RESEARCH                            INDIVIDUAL
                                                REGULAR
                                                COMMENTS =
                                           AREA = SOCIAL STUDIES           NO =  1    9/28/74  HILLIARD
                                                ORIENTATION                                         INDIVIDUAL
                                                REGULAR
                                                COMMENTS =     INTERESTING
                                                               STIMULATING

                             L.A. SUMMARY  AREA = ART                          COMPLETED  40% OF  5 ACTIVITIES
                                                                                          MISSING  3 ACTIVITIES
                                           AREA = CURRICULUM                   COMPLETED   8% OF 12 ACTIVITIES
                                                                                          MISSING 11 ACTIVITIES
                                           AREA = HUMAN GROWTH AND DEV.        COMPLETED   9% OF 11 ACTIVITIES
                                                                                          MISSING 10 ACTIVITIES
                                           AREA = LANGUAGE ARTS                COMPLETED  11% OF  9 ACTIVITIES
                                                                                          MISSING  8 ACTIVITIES
                                           AREA = MATH                         COMPLETED  13% OF  8 ACTIVITIES
                                                                                          MISSING  7 ACTIVITIES
                                           AREA = MUSIC                        COMPLETED  20% OF  5 ACTIVITIES
                                                                                          MISSING  4 ACTIVITIES
                                           AREA = READING                      COMPLETED  22% OF  9 ACTIVITIES
                                                                                          MISSING  7 ACTIVITIES
                                           AREA = SCIENCE                      COMPLETED  13% OF 15 ACTIVITIES
                                                                                          MISSING 13 ACTIVITIES
                                           AREA = SOCIAL FOUNDATIONS           COMPLETED  50% OF  4 ACTIVITIES
                                                                                          MISSING  2 ACTIVITIES
                                           AREA = SOCIAL STUDIES               COMPLETED  14% OF  7 ACTIVITIES
                                                                                          MISSING  6 ACTIVITIES
                                           ALL LEARNING ACTIVITIES                  COMPLETED  16% OF 88 ACTIVITIES
                                                                                          MISSING 74 ACTIVITIES

                             FIELD         YR = 74   QTR = 1   HAWTHORNE     BLUE
                                                GRADE = THIRD                  OPEN
                                                ROLE =         INDIVIDUAL INSTRUCTION
                                                               SMALL GROUP INSTRUCTION
                                                STRENGTHS =    AWARE OF INDIVIDUAL NEEDS
                                                               AWARE OF INDIVIDUAL PROBLEMS
                                                               GOOD WITH CHILDREN
                                                               SENSITIVE TO STUDENTS FEELINGS
                                                NEEDS =        TO PREPARE UNITS
                                                               SMALL GROUP EXPERIENCE
                                                               ART EXPERIENCE
                                                               P.E. EXPERIENCE

                             FIELD SUMMARY HAWTHORNE     GRADE = THIRD                 OPEN

                             COMMUNITY     YR = 75   QTR = 2   GATOR NUSERY            AGENCY REP =    DOE
                                           ASSISTANT
                                                ROLE =         GENERAL NURSERY ACTIVITIES
                                                               ORGANIZED GAMES
                                                               PLAYED WITH CHILDREN
                                                COMMENTS =     INITIATIVE
                                                               WORKS WELL WITH CHILDREN
```

PURGE

Output Description	Deletes records of non-registered students from the main file.
Class	S
Tape Drives	3
DD Statements	OLD NEW DELETE LRECL=3693,BLKSIZE=3693

Data Setup

Card	Contents	Column
1	Date (e.g. 1/13/75)	1-8
One card for each non-registered student	Social security number of the non-registered student	1-9

Non-registered students to be eliminated from the main file must be listed in ascending order by social security number.

NON-REGISTERED STUDENTS

Team Number: ____________________

Date: __________________________

Directions: Please list in ascending order by social security number the name and associated social security number of all students who are not registered this quarter.

Name	Social security number

SAMPLE DATA DECK FOR PURGE

```
COLUMN     0        1         2         3         4         5         6         7         8
NUMBER     12345678901234567890123456789012345678901234567890123456789012345678901234567890

           1/13/75
           100975253
           111111111
           159533426
           229511687
```

```
1/13/75
PURGE
OUTPUT DESCRIPTION: ASCENDING LIST OF STUDENTS WHOSE RECORDS WERE PURGED
     FROM THE MAIN FILE AND SAVED UNLESS OTHERWISE INDICATED


ERROR_MESSAGE         S.S.NUMBER            NAME
                      100975253             SHOWERS,APRIL
MISSING               111111111
                      150533426             FLORA,ROSE
                      229511687             PEPPERCORN,CORNELIUS
```

REQUEST

Output Description	Prints personal information on those students requesting field experience.
Class	S
Tape Drives	1
DD Statements	OLD

Data Setup

Card	Contents	Column
1	Lowest team number to be printed	1
	Highest team number to be printed	6
2	Date (e.g. 1/13/75)	1-8
3	Request date (e.g. Spring 1975) for the upcoming field experience	1-15
4	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47

SAMPLE DATA DECK FOR REQUEST

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890


          3    3
          1/13/75
          WINTER 1975
           1
          3    1    1     8
```

```
1/13/75
REQUEST
OUTPUT DESCRIPTION: LISTING OF PERSONAL INFORMATION ON STUDENTS REQUESTING FIELD EXPERIENCE
REQUEST FOR WINTER 1975


NAME:                   MERRI BEANE

LOCAL ADDRESS:          77 SUNSET DRIVE                             HOME ADDRESS:           1976 HIGH STREET #2
                        SUNRISE, FL  33313                                                  SUNLAND, CA  91040
                        3765935                                                             7146539117

DATE OF BIRTH:           2/29/53

TRAVEL EXPERIENCE:      MEXICO                                      WORK EXPERIENCE:        BABYSITTER
                        U.S.A.- ALL OVER                                                    CAMP COUNSELOR
                        CARIBBEAN                                                           RECREATION CENTER
                                                                                            NURSERY SCHOOL

CURRENT EMPLOYMENT:     CITY OF FROSTPROOF                          HOURS PER WEEK =   20

SEMINAR LEADER:         KINZER

SPECIALIZATION:         ELEMENTARY

FIELD EXPERIENCES
COMPLETED TO DATE:      YR =74   QTR =1   LEVEL =
                            SCHOOL =   HAWTHORNE         TEACHER =   BLUE               GRADE =   THIRD
                            ROLE =           INDIVIDUAL INSTRUCTION
                                             SMALL GROUP INSTRUCTION
                            STRENGTHS =      AWARE OF INDIVIDUAL NEEDS
                                             AWARE OF INDIVIDUAL PROBLEMS
                                             GOOD WITH CHILDREN
                                             SENSITIVE TO STUDENTS FEELINGS
                            NEEDS =          TO PREPARE UNITS
                                             SMALL GROUP EXPERIENCE
                                             ART EXPERIENCE
                                             P.E. EXPERIENCE
```

SS

Output Description	Prints an alphabetical list of team members and their social security numbers by seminar.
Class	S
Tape Drives	1
DD Statements	OLD

Data Setup

Card	Contents	Column
1	Lowest team number to be printed	1
	Highest team number to be printed	6
2	Date (e.g. 1/13/75)	1-8
3	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47

SAMPLE DATA DECK FOR SS

```
COLUMN     0        1         2         3         4         5         6         7         8
NUMBER     12345678901234567890123456789012345578901234567890123456789012345678901234567890

           3    3
           1/13/75
            1
           3    1    1     8
```

```
1/13/75
SS
OUTPUT DESCRIPTION: ALPHABETICAL LIST OF TEAM MEMBERS AND THEIR SOCIAL SECURITY NUMBERS BY SEMINAR
TEAM    3



SINZER________


292927404       ADAMS,EVE
543947429       BABBLE,BROOKE
195459930       BEACH,SANDI
965973517       BEANE,MERRI
652697760       BENZ,MERCEDES
737464575       BOSS,BRENDA
113422173       DARLING,DAVID
645323665       DRIGGERS,DRUSILLA
335113115       FANG,FAYE
501571614       FERN,FRANCES
159533426       FLORA,ROSE
635733213       GLORY,VICTORIA
809590110       GRUBE,OLGA
660657471       GRUMP,GLENDA
232030256       HOOD,ROBIN
375720480       HORNE,HILDA
121717553       JOLLEY,HOLLY
350303360       KEAN,ELIZABETH
765201353       KLINKER,CLARIBEL
346735487       LIES,LILY
128073997       LOBO,LULU
353034261       LOST,LORNA
229511687       PEPPERCORN,CORNELIUS
100975253       SHOWERS,APRIL
340935159       SMART,SALLY
310601060       SWINDLER,JACKIE
206361040       TOPPER,TERRI
240052403       WACK,WILLA
340727685       ZEPPER,ZILDA
```

TEAM

Output Description	Prints an alphabetical list of team members by seminar.
Class	S
Tape Drives	1
DD Statements	OLD

Data Setup

Card	Contents	Column
1	Lowest team number to be printed	1
	Highest team number to be printed	6
2	Date (e.g. 1/13/75)	1-8
3	Number of teams to be printed	1-2
One card for each team	Team number	1
	Number of seminars to be printed	6
	First seminar leader:	
	Seminar number	11
	Faculty code number	16-17
	Second seminar leader:	
	Seminar number	21
	Faculty code number	26-27
	Third seminar leader:	
	Seminar number	31
	Faculty code number	36-37
	Fourth seminar leader:	
	Seminar number	41
	Faculty code number	46-47

SAMPLE DATA DECK FOR TEAM

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890

          3    3
          1/13/75
           1
          3    1    1     8
```

```
1/13/75
TEAM
OUTPUT DESCRIPTION: ALPHABETICAL LIST OF TEAM MEMBERS BY SEMINAR
TEAM    3


          KINZER_________

          ADAMS,EVE
          BABBLE,BROOKE
          BEACH,SANDI
          BEANE,MERRI
          BENZ,MERCEDES
          BOSS,BRENDA
          DARLING,DAVID
          DRIGGERS,DRUS ILLA
          FANG,FAYE
          FERN,FRANCES
          FLORA,ROSE
          GLORY,VICTORIA
          GRUBB,OLGA
          GRUMP,GRENDA
          HOOD,ROBIN
          HORNE,HILDA
          JOLLEY,HOLLY
          KEAN,ELIZABETH
          KLINKER,CLARIBEL
          LIES,LILY
          LOBO,LULU
          LOST,LORNA
          PEPPERCORN,CORNEL IUS
          SHOWERS,APRIL
          SMART,SALLY
          SWINDLER,JACKIE
          TOPPER,TERRI
          WACK,WILLA
          ZEPPER,ZILDA
```

UPDATE

Output Description	Updates information in records contained in the main file.
Class	M
Tape Drives	2
DD Statements	OLD NEW

Data Setup

Card	Contents	Column
1	Date (e.g. 1/13/75)	1-8
One set for each student record being updated		
1	Social security number	1-9
. . . Updates . . .		
Last card of the set	Number ninety-nine (trailer label)	1-2

General Information on UPDATE

Since it is vital for information in the main file to be kept up-to-date, any changes of information on students should be entered into the system as soon as possible.

Update information for the main file can be obtained on a regular basis from various sources. Included among these sources are the following forms and corrected printouts:

1. Corrected output from BASIC
2. Corrected output from REQUEST
3. Learning activity slips
4. Registration information
5. Field placement information

If any information on a student changes, she (he) should inform her (his) seminar leader. For instance:

If a student goes through drop/add, she (he) should inform her (his) seminar leader of changes to her (his) registration.

If a student becomes employed she (he) should inform her (his) seminar leader of the employer and hours worked per week.

Update Numbers

UPDATE is designed to change information contained in specific student records within the main file. An update number is used to indicate which data item or items are to be altered. The structure, PERSONAL, shown on the following page lists the update number next to that which it updates. If the update number appears next to an elementary data item, then only this item is updated. If the update number appears next to a minor structure name (e.g. NAME) then all of the elementary data items in this minor structure (e.g. LAST, FIRST, and MIDDLE) will be updated. When a minor structure is updated all information to be retained must be placed on the update card, otherwise the numeric values will revert to zero and the character strings to blank. If the update number appears next to an array name, check the corresponding description of the contents of the data card to see whether the whole array or a single item of the array is to be updated. If a whole array is to be updated, then all information to be retained must be included on the update card.

```
STRUCTURE                                            UPDATE
                                                     NUMBER

1 PERSONAL,
  2 SOC_SEC_NO CHARACTER(9),                              1
  2 NAME,                                                 2
    3 LAST CHARACTER(14),                                 3
    3 FIRST CHARACTER(10),                                4
    3 MIDDLE CHARACTER(14),                               5
  2 PERM_ADDR,                                            6
    3 STREET CHARACTER(20),                               7
    3 CITY CHARACTER(20),                                 8
    3 STATE CHARACTER(2),                                 9
    3 ZIP CHARACTER(5),                                  10
  2 PERM_PHONE CHARACTER(10),                            11
  2 LOCAL_ADDR,                                          12
    3 STREET CHARACTER(20),                              13
    3 CITY CHARACTER(20),                                14
    3 STATE CHARACTER(2),                                15
    3 ZIP CHARACTER(5),                                  16
  2 LOCAL_PHONE CHARACTER(7),                            17
  2 SEX CHARACTER(1),                                    18
  2 BIRTH_DATE,                                          19
    3 MONTH FIXED BINARY,                                20
    3 DAY FIXED BINARY,                                  21
    3 YR FIXED BINARY,                                   22
  2 MARITAL_STAT FIXED BINARY,                           23
  2 DEP_CHILDREN FIXED BINARY,                           24
  2 AGE_RANGE(2) FIXED BINARY,                           25
  2 WORK_EXP(6) FIXED BINARY,                            26
                                                         27
  2 TRAVEL_EXP(6) FIXED BINARY,                          28
                                                         29
  2 CUR_EMPLOYMENT,                                      30
    3 EMPLOYER CHARACTER(20),                            31
    3 HRS_PER_WK FIXED BINARY,                           32
  2 COL_ATT(4),                                          33
    3 INSTITUTION CHAR(30),                              34
    3 DATES,                                             35
      4 BEGIN CHARACTER(6),                              36
      4 END CHARACTER(6),                                37
  2 GPA FIXED BINARY,                                    38
  2 MYERS_BRIGGS FIXED BINARY,                           39
  2 CEP_ENTRANCE,                                        40
    3 YR FIXED BINARY,                                   41
    3 QTR FIXED BINARY,                                  42
    3 HRS_COM FIXED DECIMAL(4,1),                        43
  2 SPECIALIZATION FIXED BINARY,                         44
  2 TEAM FIXED BINARY,                                   45
  2 SEMINAR FIXED BINARY,                                46
  2 EXP_GRAD_DATE,                                       47
    3 YR FIXED BINARY,                                   48
    3 QTR FIXED BINARY,                                  49
  2 REQ_COURSE_GRADE(9) CHARACTER(1),                    50
  2 REGISTRATION(8),                                     51
    3 DATE,                                              52
      4 YR FIXED BINARY,                                 53
      4 QTR FIXED BINARY,                                54
    3 COURSES(5),                                        55
      4 COURSE_NO CHARACTER(6),                          56
      4 HRS FIXED BINARY,                                57
      4 GRADE CHARACTER(1),                              58
  2 LEARNING_A(98),                                      59
    3 FACULTY FIXED BINARY,                              60
    3 DATE CHARACTER(6),                                 61
    3 COMMENT(6) FIXED BINARY,                           62
    3 ACTIVITY_TITLE FIXED BINARY;                       63
    3 GROUP BIT(1),                                      64
    3 SUBSTITUTE BIT(1),                                 65
  2 FIELD(7),                                            66
```

```
     3 SCHOOL FIXED BINARY,
     3 GRADE FIXED BINARY,
     3 LEVEL FIXED BINARY,
     3 DATE,
       4 YR FIXED BINARY,
       4 QTR FIXED BINARY,
     3 CLASSROOM FIXED BINARY,
     3 TEACHER CHARACTER(14),
     3 F_ROLE(6) FIXED BINARY,
     3 STRENGTHS(6) FIXED BINARY,
     3 WEAKNESSES(6) FIXED BINARY,
   2 COMMUNITY,
     3 AGENCY CHARACTER(20),
     3 AGENCY_REP CHARACTER(14),
     3 LEVEL FIXED BINARY,
     3 DATE,
       4 YR FIXED BINARY,
       4 QTR FIXED BINARY,
     3 C_ROLE(4) FIXED BINARY,
     3 C_COMMENT(4) FIXED BINARY;
```

For each item to be updated, the update number is in columns 1-2. The new data which follows the update number must be coded according to the specified format.

Update Number	Card	Contents	Column
1	1	Number one	1-2
		Social security number	6-14
2	1	Number two	1-2
		Name:	
		Last	6-19
		First	21-30
		Middle	36-49
3	1	Number three	1-2
		Last name	6-19
4	1	Number four	1-2
		First name	6-15
5	1	Number five	1-2
		Middle name	6-19
6	1	Number six	1-2
		Permanent address:	
		Street	6-25
		City	31-50
		State	56-57
		Zip code	61-65
7	1	Number seven	1-2
		Street in permanent address	6-25
8	1	Number eight	1-2
		City in permanent address	6-25
9	1	Number nine	1-2
		State in permanent address	6-7
10	1	Number ten	1-2
		Zip code in permanent address	6-10
11	1	Number eleven	1-2
		Permanent phone number	6-15
12	1	Number twelve	1-2
		Local address:	
		Street	6-25
		City	31-50
		State	56-57
		Zip code	61-65

Update Number	Card	Contents	Column
13	1	Number thirteen	1-2
		Street in local address	6-25
14	1	Number fourteen	1-2
		City in local address	6-25
15	1	Number fifteen	1-2
		State in local address	6-7
16	1	Number sixteen	1-2
		Zip code in local address	6-10
17	1	Number seventeen	1-2
		Local phone number	6-12
18	1	Number eighteen	1-2
		Sex code	6
19	1	Number nineteen	1-2
		Birthdate:	
		Month number	6-7
		Day	11-12
		Year (last two digits)	16-17
20	1	Number twenty	1-2
		Month number in birthdate	6-7
21	1	Number twenty-one	1-2
		Day in birthdate	6-7
22	1	Number twenty-two	1-2
		Year (last two digits) in birthdate	6-7
23	1	Number twenty-three	1-2
		Marital status code number	6
24	1	Number twenty-four	1-2
		Number of dependent children	6
25	1	Number twenty-five	1-2
		Age range of the dependent children:	
		Youngest	6-7
		Oldest	11-12
26	1	Number twenty-six	1-2
		First work experience code number	6-7
		Second work experience code number	11-12
		Third work experience code number	16-17
		Fourth work experience code number	21-22
		Fifth work experience code number	26-27
		Sixth work experience code number	31-32

Update Number	Card	Contents	Column
27	1	Number twenty-seven	1-2
		Add a new work experience code number	6-7
28	1	Number twenty-eight	1-2
		First travel experience code number	6-7
		Second travel experience code number	11-12
		Third travel experience code number	16-17
		Fourth travel experience code number	21-22
		Fifth travel experience code number	26-27
		Sixth travel experience code number	31-32
29	1	Number twenty-nine	1-2
		Add a new travel experience code number	6-7
30	1	Number thirty	1-2
		Current employment:	
		Employer	6-25
		Hours worked per week	31-32
31	1	Number thirty-one	1-2
		Employer in current employment	6-25
32	1	Number thirty-two	1-2
		Hours worked per week in current employment	6-7
33	1	Number thirty-three	1-2
		Number of the college attended	6-7
		College name	11-40
		Attendance dates:	
		Beginning (e.g. 74 F)	46-51
		Ending	56-61
34	1	Number thirty-four	1-2
		Number of the college attended	6-7
		College name	11-40
35	1	Number thirty-five	1-2
		Number of the college attended	6-7
		Attendance dates:	
		Beginning (e.g. 74 F)	11-16
		Ending	21-26
36	1	Number thirty-six	1-2
		Number of the college attended	6-7
		Beginning attendance date (e.g. 74 F)	11-16
37	1	Number thirty-seven	1-2
		Number of the college attended	6-7
		Ending attendance date (e.g. 74 F)	11-16

Update Number	Card	Contents	Column
38	1	Number thirty-eight	1-2
		GPA code number	6
39	1	Number thirty-nine	1-2
		Myers Briggs type code number	6-7
40	1	Number forty	1-2
		CEP Entrance date:	
		Year (last two digits)	6-7
		Quarter code number	11
		Hours entering program	16-19
41	1	Number forty-one	1-2
		Year (last two digits) in CEP entrance date	6-7
42	1	Number forty-two	1-2
		Quarter code number in CEP entrance date	6
43	1	Number forty-three	1-2
		Hours entering program	6-9
44	1	Number forty-four	1-2
		Specialization code number	6
45	1	Number forty-five	1-2
		Team number	6
46	1	Number forty-six	1-2
		Seminar number	6
47	1	Number forty-seven	1-2
		Expected graduation date:	
		Year (last two digits)	6-7
		Quarter code number	11
48	1	Number forty-eight	1-2
		Year (last two digits) in expected graduation date	6-7
49	1	Number forty-nine	1-2
		Quarter code number in expected graduation date	6
50	1	Number fifty	1-2
		Required course code number	6-7
		Grade	11

Update Number	Card	Contents	Column
51	1	Number fifty-one	1-2
		Registration number (quarter in CEP)	6-7
		Registration date:	
		Year (last two digits)	11-12
		Quarter code number	16
		Courses taken:	
		First course:	
		Course number	21-26
		Credit hours	31-32
		Grade	36
		Second course:	
		Course number	41-46
		Credit hours	51-52
		Grade	56
		Third course:	
		Course number	61-66
		Credit hours	71-72
		Grade	76
	2	Fourth course:	
		Course number	6-11
		Credit hours	16-17
		Grade	21
		Fifth course:	
		Course number	26-31
		Credit hours	36-37
		Grade	41
52	1	Number fifty-two	1-2
		Registration number (quarter in CEP)	6-7
		Registration date:	
		Year (last two digits)	11-12
		Quarter code number	16
53	1	Number fifty-three	1-2
		Registration number (quarter in CEP)	6-7
		Year (last two digits) in registration date	11-12
54	1	Number fifty-four	1-2
		Registration number (quarter in CEP)	6-7
		Quarter code number in registration date	11
55	1	Number fifty-five	1-2
		Registration number (quarter in CEP)	6-7
		Courses taken:	
		First course:	
		Course number	11-16
		Credit hours	21-22
		Grade	26

Update Number	Card	Contents	Column
		Second course:	
		Course number	31-36
		Credit hours	41-42
		Grade	46
		Third course:	
		Course number	51-56
		Credit hours	61-62
		Grade	66
	2	Fourth course:	
		Course number	6-11
		Credit hours	16-17
		Grade	21
		Fifth course:	
		Course number	26-31
		Credit hours	36-37
		Grade	41
56	1	Number fifty-six	1-2
		Registration number (quarter in CEP)	6-7
		Course entry number	11-12
		Course number	16-21
57	1	Number fifty-seven	1-2
		Registration number (quarter in CEP)	6-7
		Course entry number	11-12
		Credit hours	16-17
58	1	Number fifty-eight	1-2
		Registration number (quarter in CEP)	6-7
		Course entry number	11-12
		Grade	16
59	1	Number fifty-nine	1-2
		Substantive area code number	6-7
		Activity number	11-12
		Faculty code number	16-17
		Completion date	21-26
		Learning activity comments:	
		First comment code number	31-32
		Second comment code number	36-37
		Third comment code number	41-42
		Fourth comment code number	46-47
		Fifth comment code number	51-52
		Sixth comment code number	56-57
		Activity title code number	61-63
		Group status code	66
		Substitute status code	71

Update Number	Card	Contents	Column
60	1	Number sixty	1-2
		Substantive area code number	6-7
		Activity number	11-12
		Faculty code number	16-17
61	1	Number sixty-one	1-2
		Substantive area code number	6-7
		Activity number	11-12
		Completion date (e.g. month, day, year 011375)	16-21
62	1	Number sixty-two	1-2
		Substantive area code number	6-7
		Activity number	11-12
		Learning activity comments:	
		First comment code number	16-17
		Second comment code number	21-22
		Third comment code number	26-27
		Fourth comment code number	31-32
		Fifth comment code number	36-37
		Sixth comment code number	41-42
63	1	Number sixty-three	1-2
		Substantive area code number	6-7
		Activity number	11-12
		Activity title code number	16-18
64	1	Number sixty-four	1-2
		Substantive area code number	6-7
		Activity number	11-12
		Group status code	16
65	1	Number sixty-five	1-2
		Substantive area code number	6-7
		Activity number	11-12
		Substitute status code	16
66	1	Number sixty-six	1-2
		Field experience number	6-7
		School code number	11-12
		Grade code number	16-17
		Level in CEP code number	21
		Date:	
		Year (last two digits)	26-27
		Quarter code number	31
		Classroom type code number	36
		Teachers name	41-54

Update Number	Card	Contents	Column
	2	Field experience roles:	
		First role code number	6-7
		Second role code number	11-12
		Third role code number	16-17
		Fourth role code number	21-22
		Fifth role code number	26-27
		Sixth role code number	31-32
		Field experience strengths:	
		First strength code number	36-37
		Second strength code number	41-42
		Third strength code number	46-47
		Fourth strength code number	51-52
		Fifth strength code number	56-57
		Sixth strength code number	61-62
	3	Field experience needs:	
		First weaknesses code number	6-7
		Second weaknesses code number	11-12
		Third weaknesses code number	16-17
		Fourth weaknesses code number	21-22
		Fifth weaknesses code number	26-27
		Sixth weaknesses code number	31-32
67	1	Number sixty-seven	1-2
		Field experience number	6-7
		School code number	11-12
68	1	Number sixty-eight	1-2
		Field experience number	6-7
		Grade code number	11-12
69	1	Number sixty-nine	1-2
		Field experience number	6-7
		Level in CEP code number	11
70	1	Number seventy	1-2
		Field experience number	6-7
		Date:	
		Year (last two digits)	11-12
		Quarter code number	16
71	1	Number seventy-one	1-2
		Field experience number	6-7
		Year (last two digits) in field experience date	11-12
72	1	Number seventy-two	1-2
		Field experience number	6-7
		Quarter code number in field experience date	11

Update Number	Card	Contents	Column
73	1	Number seventy-three	1-2
		Field experience number	6-7
		Classroom type code number	11-12
74	1	Number seventy-four	1-2
		Field experience number	6-7
		Teachers name	11-24
75	1	Number seventy-five	1-2
		Field experience number	6-7
		Field experience roles:	
		First role code number	11-12
		Second role code number	16-17
		Third role code number	21-22
		Fourth role code number	26-27
		Fifth role code number	31-32
		Sixth role code number	36-37
76	1	Number seventy-six	1-2
		Field experience number	6-7
		Field experience strengths:	
		First strength code number	11-12
		Second strength code number	16-17
		Third strength code number	21-22
		Fourth strength code number	26-27
		Fifth strength code number	31-32
		Sixth strength code number	36-37
77	1	Number seventy-seven	1-2
		Field experience number	6-7
		Field experience needs:	
		First weaknesses code number	11-12
		Second weaknesses code number	16-17
		Third weaknesses code number	21-22
		Fourth weaknesses code number	26-27
		Fifth weaknesses code number	31-32
		Sixth weaknesses code number	36-37
78	1	Number seventy-eight	1-2
		Agency name	6-25
		Agency representative	31-44
		Level in CEP code number	46
		Date:	
		Year (last two digits)	51-52
		Quarter code number	56
	2	Community experience roles:	
		First role code number	6-7
		Second role code number	11-12
		Third role code number	16-17
		Fourth role code number	21-22

Update Number	Card	Contents	Column
		Community experience comments:	
		First comment code number]	26-27
		Second comment code number	31-32
		Third comment code number	36-37
		Fourth comment code number	41-42
79	1	Number seventy-nine	1-2
		Agency name	6-25
80	1	Number eighty	1-2
		Agency representative	6-19
81	1	Number eighty-one	1-2
		Level in CEP code number	6
82	1	Number eighty-two	1-2
		Date in community experience:	
		Year (last two digits)	6-7
		Quarter code number	11
83	1	Number eighty-three	1-2
		Year (last two digits) in community experience date	6-7
84	1	Number eighty-four	1-2
		Quarter code number in community experience date	6
85	1	Number eighty-five	1-2
		Community experience roles:	
		First role code number	6-7
		Second role code number	11-12
		Third role code number	16-17
		Fourth role code number	21-22
86	1	Number eighty-six	1-2
		Community experience comments:	
		First comment code number	6-7
		Second comment code number	11-12
		Third comment code number	16-17
		Fourth comment code number	21-22

COMMUNITY EXPERIENCE PLACEMENT FORM

Social security number: ______________________

Name:
Last: ______________________
First: ______________________
Middle: ______________________

Agency name: ______________________

Agency representative's name (first initial and last name):

____

Level in CEP:

Please check your level in CEP.

Initiate	_____(1)
Assistant	_____(2)
Associate	_____(3)
Intensive	_____(4)

Community experience date:
Year (last two digits):______________
Quarter:

Please check the quarter.

Fall	_____(1)
Winter	_____(2)
Spring	_____(3)
Summer	_____(4)

FIELD PLACEMENT FORM

Social security number: ______________________________

Name:
Last: ____________________________
First: ____________________________
Middle: ____________________________

Quarter number:

Please check how many quarters you have been placed in local schools.

One ______(1)
Two ______(2)
Three ______(3)
Four ______(4)
Five ______(5)
Six ______(6)
Seven ______(7)

School:

Please check the name of the school in which you have been placed.

J. J. Finley ______(1)
Hawthorne ______(2)
Sidney Lanier ______(3)
Littlewood ______(4)
Shell ______(5)
Tewilliger ______(6)
Westwood ______(7)

Grade:

Please check the grade level at which you have been placed.

First ______(1)
Second ______(2)
Third ______(3)
Fourth ______(4)
Fifth ______(5)
Sixth ______(6)
Seventh ______(7)
Eighth ______(8)
Pre-kindergarten ______(9)
Kindergarten ______(10)
Multi-age primary ______(11)
Multi-age upper elementary ______(12)
Middle school ______(13)
Special education ______(14)
Music ______(15)
Reading ______(16)
P.E. ______(17)

Level in CEP:

Please check your level in CEP.

Initiate ______(1)
Assistant ______(2)
Associate ______(3)
Intensive ______(4)

Field experience date:
Year (last two digits): ____________________
Quarter: Please check the quarter.
Fall ____(1)
Winter ____(2)
Spring ____(3)
Summer ____(4)

Classroom type: Please check the type of classroom in which you have been placed.
Basic skills ____(1)
Free ____(2)
Open ____(3)
Standard ____(4)

Directing teacher's name (first initial and last name):

____

FIELD EXPERIENCE EVALUATION

(PROSPECTIVE TEACHER)	(LEVEL)
(SCHOOL)	(GRADE LEVEL OR AREA)
(TEACHER)	(DATE)

(DIRECTING TEACHERS SIGNATURE)

PLEASE CHECK UP TO SIX ITEMS BELOW WHICH DESCRIBE THIS PROSPECTIVE TEACHERS ROLE(OPPORTUNITIES AND REPONSIBILITIES) WHILE TEACHING WITH YOU THIS PAST QUARTER.

ADMINISTERED SOCIO-GRAM ---- (1)
ADMINISTERED STANDARDIZED TEST ---- (2)
ADMINISTERED TESTS ---- (3)
ARRANGED CLASSROOM ENVIORNMENT ---- (4)
ARRANGED SCHOOL VISIT FROM A RESC ---- (5)
ATTENDED SCHOOL FUNCTION ---- (6)
DEVELOPED LEARNING CENTERS ---- (7)
DIRECTED PRODUCTION OF A PLAY OR ---- (8)
DIRECTED SHARING ACTIVITIES ---- (9)
ENRICHMENT ACTIVITIES ---- (10)
EVALUATED PAPERS ---- (11)
EVALUATED STUDENT PROGRESS ---- (12)
GEN. ACTIVITIES AND PROCEDURES ---- (13)
INDIVIDUAL INSTRUCTION ---- (14)
LARGE GROUP INSTRUCTION ---- (15)
LEAD CLASSROOM DISCUSSION ---- (16)
MADE HOME VISIT ---- (17)
MADE MATERIALS AND/OR DISPLAYS ---- (18)
OBSERVED CLASS ---- (19)
OBSERVED PSYCH SERVICES ---- (20)
OBSERVED TEAM TEACHING ---- (21)
PARENT TEACHER CONFERENCE ---- (22)
PLANNED ACTIVITIES AROUND UNITS ---- (23)
PLANNED AND SUPERVISED ACTIVITIES ---- (24)
PLANNED GAMES AND ACTIVITIES ---- (25)
PLANNED GROUP INSTRUCTION ---- (26)
PLANNED WORK TIME ---- (27)
PLAYED GAMES WITH CHILDREN ---- (28)
PREPARED INDEPENDENT WORK ---- (29)
PREPARED LESSONS FOR CLASS ---- (30)
READ STORIES TO CLASS ---- (31)
SMALL GROUP INSTRUCTION ---- (32)
SOME TEAM TEACHING ---- (33)
TAUGHT ART ---- (34)
TAUGHT LANGUAGE ARTS ---- (35)
TAUGHT MATH ---- (36)
TAUGHT MUSIC ---- (37)
TAUGHT P.E. ---- (38)
TAUGHT SCIENCE ---- (39)
TAUGHT SOCIAL STUDIES ---- (40)
TAUGHT READING ---- (41)
TEAM PLANNING ---- (42)
SMALL GROUP INSTRUCTION ---- (43)
USED AUDIO-VISUAL EQUIPMENT ---- (44)
WORKED IN ALL AREAS ---- (45)
WORKED WITH EMR CHILDREN ---- (46)
WORKED WITH FOREIGN STUDENTS ---- (47)
WORKED WITH GROUPS ---- (48)
WORKED WITH LEARNING CENTERS ---- (49)

PLEASE CHECK UP TO SIX ITEMS BELOW WHICH CHARACTERIZE THIS PROSPECTIVE TEACHERS STRENGTHS.

ADAPTS MATERIALS TO ACHIEVEMENT LEVEL ____ (1)
AWARE OF INDIVIDUAL INTERESTS ____ (2)
AWARE OF INDIVIDUAL NEEDS ____ (3)
AWARE OF INDIVIDUAL PROBLEMS ____ (4)
CAPABLE ____ (5)
CHILDREN LIKE HER(HIM) ____ (6)
CONCERNED WITH CHILDS WELFARE ____ (7)
CONFIDENT ____ (8)
CONSTRUCTS GOOD EVALUATION INSTRUMENT ____ (9)
COOPERATIVE ____ (10)
CREATIVE ____ (11)
DEPENDABLE ____ (12)
DESIRE TO HELP CHILDREN ____ (13)
DESIRE TO LEARN ____ (14)
EFFECTIVE CLASSROOM CONTROL ____ (15)
EMPHASIZES INDIVIDUAL NEEDS ____ (16)
EVALUATES ACCURATELY ____ (17)
EXCELLENT PLANNER ____ (18)
EXCELLENT RAPPORT ____ (19)
EXCELLENT RELATION WITH STAFF ____ (20)
EXCELLENT RELATION WITH PARENTS ____ (21)
EXCEPTIONALLY CAPABLE IN ART ____ (22)
EXECPTIONALLY CAPABLE IN LANG.A. ____ (23)
EXCEPTIONALLY CAPABLE IN MATH ____ (24)
EXCEPTIONALLY CAPABLE IN MUSIC ____ (25)
EXCEPTIONALLY CAPABLE IN READING ____ (26)
EXCEPTIONALLY CAPABLE IN SCIENCE ____ (27)
EXECPTIONALLY CAPABLE IN S.S. ____ (28)
FLEXIBLE ____ (29)
GAINS STUDENTS TRUST ____ (30)
GOES BEYOND WHAT IS EXPECTED ____ (31)
GOOD RAPPORT ____ (32)
GOOD VOICE CONTROL ____ (33)
GOOD WITH CHILDREN ____ (34)
GROWTH IN DEPENDIBILITY ____ (35)
HELPS STUDENTS FEEL SUCCESSFUL ____ (36)
INNOVATIVE ____ (37)
INTEREST IN CHILDRENS WORK ____ (38)
INITIATES CLASS INVOLVEMENT ____ (39)
KNOWLEDGEABLE ____ (40)
LARGE GROUP WORK ____ (41)
LIKES THIS AGE GROUP ____ (42)
LIKES WORKING WITH CHILDREN ____ (43)
MOTIVATES AND INTERESTS STUDENTS ____ (44)
PATIENT ____ (45)
PERSERVERENCE ____ (46)
PLANS EFFECTIVELY ____ (47)
POSITIVE ATTITUDE ____ (48)
PROFESSIONAL PROMISE ____ (49)
PROMPT ____ (50)
RESPONDS WELL TO CRITICISM ____ (51)
SEES WHAT NEEDS TO BE DONE ____ (52)
SENSE OF RESPONSIBILITY ____ (53)
SENSITIVE TO STUDENTS FEELINGS ____ (54)
SHOWS INITIATIVE ____ (55)
SMALL GROUP WORK ____ (56)
SYMPATHETIC ____ (57)
TEACHES FROM CHILDRENS INTERESTS ____ (58)
TREATS CHILDREN FAIRLY ____ (59)
UNDERSTANDS CHILDREN ____ (60)
USES INDIVIDUAL DIFFERENCES ____ (61)
USES OUTSIDE RESOURCES WELL ____ (62)
USES PEER TEACHING ____ (63)
VERY RESPONSIBLE ____ (64)
WANTS CHILDREN TO UNDERSTAND CONCEPTS ____ (65)
WELL ORGANIZED ____ (66)
WELL PREPARED ____ (67)
WORK WITH LEARNING PROBLEMS ____ (68)

PLEASE CHECK UP TO SIX ITEMS BELOW WHICH IDENTIFY THIS PROSPECTIVE TEACHERS AREAS OF WEAKNESSES AND/OR NEEDED IMPROVEMENT

ART EXPERIENCE ____ (1)
BETTER RAPPORT WITH STUDENTS ____ (2)
BETTER RAPPORT WITH STAFF ____ (3)
BETTER RAPPORT WITH PARENTS ____ (4)
BETTER UNDERSTANDING OF CHILDREN ____ (5)
BETTER VOICE CONTROL ____ (6)
CHALKBOARD EXPERIENCE ____ (7)
CONCERN FOR CHILDRENS WELFARE ____ (8)
DISCIPLINE EXPERIENCE ____ (9)
EVALUATION EXPERIENCE ____ (10)
EXPERIENCE DIRECTING LESSONS ____ (11)
EXPOSURE TO LOWER GRADES ____ (12)
EXPOSURE TO MORE GRADES ____ (13)
EXPOSURE TO UPPER GRADES ____ (14)
GROUP MANAGEMENT EXPERIENCE ____ (15)
LANGUAGE ARTS EXPERIENCE ____ (16)
LARGE GROUP EXPERIENCE ____ (17)
LOWER INCOME EXPERIENCE ____ (18)
MATH EXPERIENCE ____ (19)
MORE ENTHUSIASM ____ (20)
MORE PERSERVERENCE ____ (21)
MORE TEACHING TECHNIQUES ____ (22)
MUSIC EXPERIENCE ____ (23)
P.E. EXPERIENCE ____ (24)
POSITIVE ATTITUDE ____ (25)
READING EXPERIENCE ____ (26)
SCIENCE EXPERIENCE ____ (27)
SMALL GROUP EXPERIENCE ____ (28)
SOCIAL STUDIES EXPERIENCE ____ (29)
TEACHING EXPERIENCE ____ (30)
TESTING EXPERIENCE ____ (31)
TO ADAPT BETTER ____ (32)
TO BE AWARE OF INDIVIDUAL INTERESTS ____ (33)
TO BE AWARE OF INDIVIDUAL PROBLEMS ____ (34)
TO BE CONFIDENT ____ (35)
TO BE CREATIVE ____ (36)
TO BE DEPENDABLE ____ (37)
TO BE INNOVATIVE ____ (38)
TO BE LESS SENSITIVE ____ (39)
TO BE ORGANIZED ____ (40)
TO BE MORE SENSITIVE ____ (41)
TO BE PATIENT ____ (42)
TO BE PREPARED ____ (43)
TO BE PUNCTUAL ____ (44)
TO BE RESPONSIBLE ____ (45)
TO BE RELAXED ____ (46)
TO BE WILLING TO HELP CHILDREN ____ (47)
TO BE WILLING TO HELP COWORKERS ____ (48)
TO COMPLETE PROJECTS ____ (49)
TO CONSTRUCT EVALUATION INSTRUMENTS ____ (50)
TO CONTROL LARGE GROUPS ____ (51)
TO EMPHASIZE INDIVIDUAL NEEDS ____ (52)
TO HELP STUDENTS FEEL SUCCESSFUL ____ (53)
TO INITIATE CLASS INVOLVEMENT ____ (54)
TO PLAN BETTER ____ (55)
TO PREPARE UNITS ____ (56)
TO RECOGNIZE INDIVIDUAL NEEDS ____ (57)
TO RELATE BETTER ____ (58)
TO RESPOND BETTER TO CRITICISM ____ (59)
TO SHOW INITIATIVE ____ (60)
TO TEACH FROM CHILDRENS INTEREST ____ (61)
TO TREAT STUDENTS FAIRLY ____ (62)
TO USE OUTSIDE RESOURCES BETTER ____ (63)
TO USE PEER TEACHING ____ (64)
WORK IN CREATIVE AREAS ____ (65)
WORK WITH LEARNING PROBLEMS ____ (66)

LEARNING ACTIVITY SLIP

Student's name: ______________________________

Student's social security number: _______________

Team number: ___________________

Seminar: ______________________

Date: _________________________

Area: _________________________

Activity #: ____________________

Activity title: ______________________________

Was this activity done with a group of other students? Yes____ No____

Was this activity a substitute for a regular activity? Yes____ No____

Please check up to six comments below which characterize the student's performance on this activity.

Accurate _____(1)
Adaptable _____(2)
Brief _____(3)
Careful use of ideas _____(4)
Command of subject _____(5)
Competent _____(6)
Complete _____(7)
Creative _____(8)
Critical independent thinker _____(9)
Descriptive _____(10)
Energetic _____(11)
Enthusiastic _____(12)
Excellent _____(13)
Excellent ideas _____(14)
Fair _____(15)
Flexible _____(16)
Good _____(17)
Good classroom applications _____(18)
Good criteria _____(19)
Imaginative _____(20)
Interesting _____(21)
Logical _____(22)
Motivated _____(23)
Needs more supportive evidence _____(24)
Needs more documentation _____(25)
Needs to be more concise _____(26)
Not enough sources _____(27)
Original ideas _____(28)
Precise _____(29)
Poor _____(30)
Reasoned inference _____(31)
Reliable _____(32)
Relates well to people _____(33)
Resourceful _____(34)
Satisfactory _____(35)
Scientific _____(36)
Scholarly _____(37)
Sequitur _____(38)
Steady pursuit of task _____(39)
Stimulating _____(40)
Systematic _____(41)
Thoughtfully applied _____(42)
Too general _____(43)
Well chosen readings _____(44)
Well discussed _____(45)
Well done _____(46)
Well researched _____(47)
Well stated _____(48)
Work competent if attentive _____(49)
Work shows humor _____(50)
Writing difficult to read _____(51)

Instructor's signature: ______________________________

REGISTRATION INFORMATION FORM

Social security number: ______________________________

Name:
Last: ______________________________
First: ______________________________
Middle: ______________________________

Registration number:

Please check the number of quarters you have registered in CEP (include this registration).

One	_____(1)
Two	_____(2)
Three	_____(3)
Four	_____(4)
Five	_____(5)
Six	_____(6)
Seven	_____(7)
Eight	_____(8)

Registration date:
Year (last two digits): ____________________
Quarter:

Please check the quarter.

Fall	_____(1)
Winter	_____(2)
Spring	_____(3)
Summer	_____(4)

Courses:

Please list the course number and associated credit hours for each course you are taking this quarter.

Course number	Credit hours
____________	____________
____________	____________
____________	____________
____________	____________
____________	____________

SAMPLE DATA DECK FOR UPDATE

```
COLUMN  0         1         2         3         4         5         6         7         8
NUMBER  12345678901234567890123456789012345678901234567890123456789012345678901234567890


        1/13/75
        192837645
         1   966973617
         2   BEAN           MARY
         3   BEANE
         4   MERRI
         5   MEAN
         6   1976 HIGH STREET         SUNNYSIDE               FL   32461
         7   1976 HIGH STREET #2
         8   SUNLAND
         9   CA
        10   91040
        11   7146539117
        12   98 SUMMER STREET         SUNSHINE                FL   33615
        13   77 SUNSET DRIVE
        14   SUNRISE
        15   FL
        16   33313
        17   3765935
        18   F
        19    4    1   54
        20   2
        21   29
        22   53
        23   2
        24   2
        25    1   3
        26    1    2   10
        27    8
        28    8   11
        29    5
        30   UNIVERSITY OF FLA        15
        31   CITY OF FROSTPROOF
        32   20
        33    2   UNIVERSITY OF FLORIDA              74 1      CUR
        34   1    MIAMI DADE JUNIOR COLLEGE
        35   1    72 F      73 SP
        36   1    72 SU
        37   1    73 W
```

SAMPLE DATA DECK FOR UPDATE (CONTINUED)

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890


          38   4
          39   13
          40   73   4    90.
          41   74
          42   1
          43   91.
          44   1
          45   3
          46   1
          47   75   2
          48   76
          49   3
          50   4    *
          51    1   74   4    EDE311     7          EDE405     2          MSC390      5

          51    2   75   2    EDE405     2          EDE322     8

          52    1   73   1
          53    1   74
          54    1   1
          55    1   EDE311    8   S    EDE405     2   S    MSC390     2   3

          56    1    4   PL262
          57    1    4    2
          58    1    4   A
          59    1    1   18   10 874                                    1  0    0
          59    1    4   12   101874     1    3    6   10   18   41     5  0    0
          59    2    1   17   11 574                                    1  0    0
          59    3    1    7   10 574                                  111  0    0
          59    4    1    4   112874                                    1  0    0
          59    5    1   13   102674                                    1  0    0
          59    6    1   14    92874                                    1  0    0
          59    7    1    4   101574                                    1  0    0
          59    7    2   23   101674     1                            130  0    0
          59    8    1   21   10 374    35                              1  1    1
          59    8    2   21   101774    35                            151  0    0
          59    9    1    2   11 474    49                              1  0    0
          59    9    2    2   112274                                  170  0    0
          59   10    1    6    92874    21   40                         1  0    0
```

SAMPLE DATA DECK FOR UPDATE (CONTINUED)

```
COLUMN    0        1         2         3         4         5         6         7         8
NUMBER    12345678901234567890123456789012345678901234567890123456789012345678901234567890

          60    8    1    3
          61    8    1   11 374
          62    8    1   101074
          63    8    1   150
          64    8    1   0
          65    8    1   0
          66    1    1   .4   2    73   4    4    BLACK

          67    1    2
          68    1    3
          69    1    1
          70    1   75   2
          71    1   74
          72    1   1
          73    1    3
          74    1   BLUE
          75    1   14   32
          76    1    3   4    34   54
          77    1   56  .28    1   24
          78   GATOR PLAY SCHOOL        GOOD           3    74   1

          79   GATOR NUSERY
          80   DOE
          81   2
          82   73   3
          83   75
          84   2
          85    5    7    9
          86    6    8
          99
```

```
1/13/75
UPDATE
OUTPUT DESCRIPTION: A LISTING BY SOCIAL SECURITY NUMBER OF THE OLD AND NEW CONTENTS OF ALL ITEMS WHICH HAVE BEEN UPDATED


SOCIAL_SECURITY_NO    ITEM    PREVIOUS_CONTENTS                            PRESENT_CONTENTS
    192837645

                        1     192837645                                    966973617
                        2     MEAN              MARY                       DEAN            MARY
                        3     BEAN                                         BLANE
                        4     MARY                                         MERRI
                        5                                                  MLAN
                        6      9 SHADY LANE           GOULDS               1976 HIGH STREET       SUNNYSIDE
                              FL  33170                                    FL  32461
                        7     1976 HIGH STREET                             1976 HIGH STREET #2
                        8     SUNNYSIDE                                    SUNLAND
                        9     FL                                           CA
                       10     32461                                        91040
                       11      904329792                                   7146539117
                       12      32 SNOW HILL           FROSTPROOF           98 SUMMER STREET       SUNSHINE
                              FL  33843                                    FL  33615
                       13     98 SUMMER STREET                             77 SUNSET DRIVE
                       14     SUNSHINE                                     SUNRISE
                       15     FL                                           FL
                       16     33615                                        33313
                       17     3788341                                      3765936
                       18     F                                            F
                       19      6  17  53                                    4   1  54
                       20      4                                            2
                       21      1                                           29
                       22     54                                           53
                       23     1                                            2
                       24     0                                            2
                       25      0   0                                        1   3
                       26      1   3  11   0   0   0                        1   2  10   0   0   0
                       27      0                                            8
                       28      7  11   0   0   0   0                        8  11   0   0   0   0
                       29      0                                            5
                       30     STATE OF FLORIDA         5                   UNIVERSITY OF FLA      15
                       31     UNIVERSITY OF FLA                            CITY OF FROSTPROOF
                       32     15                                           20
                       33                                                  UNIVERSITY OF FLORIDA           74 1    CUR
```

```
14.3479
UPDATE
OUTPUT DESCRIPTION: A LISTING BY SOCIAL SECURITY NUMBER OF THE OLD AND NEW CONTENTS OF ALL ITEMS WHICH HAVE BEEN UPDATED

SOCIAL SECURITY NO         ITEM    PREVIOUS CONTENTS                                   PRESENT CONTENTS
                            NO     VALENCIA COMMUNITY COLLEGE                          MIAMI DADE JUNIOR COLLEGE
                           35      71 F     73 W                                       72 F       73 SP
                           36      72 F                                                72 SU
                           37      73 SP                                               73 W
                           38      3                                                   4
                           39      11                                                  13
                           40      74    1    0.0                                      73   4    90.0
                           41      73                                                  74
                           42      4                                                   1
                           43      90.0                                                91.0
                           44      2                                                   1
                           45      2                                                   3
                           46      2                                                   1
                           47      75    3                                             75   2
                           48      75                                                  76
                           49      2                                                   3
                           50                             *                            *
                           51      0   0                                               74   4
                                             0               0               0         EDE311     7    EDE405    2     MSC390    5
                                             0               0                                  0              0
                           51      0   0                                               75   2
                                             0               0               0         EDE405     2    EDE322    8               0
                                             0               0                                  0              0
                           52      74   4                                              73   1
                           53      73                                                  74
                           54      1                                                   1
                           55      EDE311    7     EDE405    2     MSC390    5         EDE311    8 S   EDE405     2  S   MSC390    2   8
                                   MSC390    5               0                         MSC390    2 B             0
                           56                                                          PL262
                           57      0                                                   2
                           58                                                          A
                           59      0           0   0   0   0   0   0    0   0   0      18  10 674   0   0   0   0   0   0    1   0   0
                           59      0           0   0   0   0   0   0    0   0   0      12  101874   1   3   0  10  18  41    5   0   0
                           59      0           0   0   0   0   0   0    0   0   0      17  11 574   0   0   0   0   0   0    1   0   0
                           59      0           0   0   0   0   0   0    0   0   0       7  10 574   0   0   0   0   0   0  111   0   0
                           59      0           0   0   0   0   0   0    0   0   0       4  112874   0   0   0   0   0   0    1   0   0
                           59      0           0   0   0   0   0   0    0   0   0      13  102674   0   0   0   0   0   0    1   0   0
                           59      0           0   0   0   0   0   0    0   0   0      14   92874   0   0   0   0   0   0    1   0   0
```

```
1/13/75
UPDATE
OUTPUT DESCRIPTION: A LISTING BY SOCIAL SECURITY NUMBER OF THE OLD AND NEW CONTENTS OF ALL ITEMS WHICH HAVE BEEN UPDATED


SOCIAL SECURITY NO    ITEM    PREVIOUS CONTENTS                             PRESENT CONTENTS
                       59     0          0  0  0  0  0  0   0  0  0    4  101574   0  0  0  0  0  0    1  0  0
                       59     0          0  0  0  0  0  0   0  0  0   23  101674   1  0  0  0  0  0  130  0  0
                       59     0          0  0  0  0  0  0   0  0  0   21  10 374  35  0  0  0  0  0    1  1  1
                       59     0          0  0  0  0  0  0   0  0  0   21  101774  35  0  0  0  0  0  151  0  0
                       59     0          0  0  0  0  0  0   0  0  0    2  11 474  49  0  0  0  0  0    1  0  0
                       59     0          0  0  0  0  0  0   0  0  0    2  112274   0  0  0  0  0  0  170  0  0
                       59     0          0  0  0  0  0  0   0  0  0    6   92874  21 40  0  0  0  0    1  0  0
                       60     21                                     3
                       61     10 374                                 11 374
                       62     35  0  0  0  0  0                      10  4  0  0  0  0
                       63      1                                     150
                       64     1                                      0
                       65     1                                      0
                       66      0  0  0  0  0  0                        1  4  2  73  4  4  BLACK
                               0  0  0  0  0  0  0  0  0  0  0  0    0  0  0  0  0  0  0  0  0  0  0  0
                               0  0  0  0  0  0                      0  0  0  0  0  0
                       67      1                                     2
                       68      4                                     3
                       69     2                                      0
                       70     73  4                                  75  2
                       71     75                                     74
                       72     2                                      1
                       73      4                                     3
                       74     BLACK                                  BLUE
                       75      0  0  0  0  0  0                      14  32  0  0  0  0
                       76      0  0  0  0  0  0                       3   4 34 54  0  0
                       77      0  0  0  0  0  0                      56  28  1 24  0  0
                       78                                  0  0  0   GATOR PLAY SCHOOL     GOOD          3  74  1
                               0  0  0  0  0  0  0  0                    0  0  0  0   0  0  0  0
                       79     GATOR PLAY SCHOOL                      GATOR NUSERY
                       80     GOOD                                   DOE
                       81     3                                      2
                       82     74  1                                  73  3
                       83     73                                     75
                       84     3                                      2
                       85      0  0  0  0                             5  7  9  0
                       86      0  0  0  0                             6  8  0  0
```

DICTIONARY FOR THE AREA CODES

CODE NUMBER	ITEM NAME
1	ART
2	CURRICULUM
3	HUMAN GROWTH AND DEV.
4	LANGUAGE ARTS
5	MATH
6	MUSIC
7	READING
8	SCIENCE
9	SOCIAL FOUNDATIONS
10	SOCIAL STUDIES
11	BEYOND MINIMUM

DICTIONARY FOR THE CLASSROOM TYPE CODES

CODE NUMBER	ITEM NAME
0	
1	BASIC SKILLS
2	FREE
3	OPEN
4	STANDARD

DICTIONARY FOR THE COMMUNITY COMMENT CODES

CODE NUMBER	ITEM NAME
0	
1	COMMON SENSE
2	COOPERATIVE
3	CREATIVE THINKING
4	ENTHUSIASTIC
5	INDUSTRIOUS
6	INITIATIVE
7	RELATES WELL WITH CHILDREN
8	WORKS WELL WITH CHILDREN
9	WORKS WELL WITH COWORKERS
10	WORKS WELL WITH SUPERVISOR

DICTIONARY FOR THE COMMUNITY EXPERIENCE ROLE CODES

CODE NUMBER	ITEM NAME
0	
1	ANSWERED PHONE
2	ASSISTED IN SPECIAL CLASSES
3	COLLECTED SAMPLES OF DISPLAY USAGE
4	CONDUCTED TOURS
5	GENERAL NURSERY ACTIVITIES
6	OBSERVED
7	ORGANIZED GAMES
8	PART TIME LIBRARIAN
9	PLAYED WITH CHILDREN
10	VISITED PATIENTS
11	WORKED WITH ARTS AND CRAFTS
12	WORKED WITH FREE PLAY ACTIVITIES
13	WORKED WITH MUSIC AND READINESS ACTIVITIES
14	WROTE ANCEDOTAL SITUIATIONS
15	

DICTIONARY FOR THE FACULTY NAME CODES

CODE NUMBER	ITEM NAME
1	BERNARD
2	CURRAN
3	DETURE
4	FEY
5	GOLDMAN
6	HILLIARD
7	JESTER
8	KINZER
9	LEWIS
10	LITCHER
11	MARTINELLO
12	NEWCOMB
13	OLSON
14	PENLAND
15	ROWE
16	SERAPHINE
17	SHEA
18	SMITH
19	TISON
20	WILMOT
21	KRALL
22	CREEDON
23	LEVINE
24	HEDGES
25	GRIMM
26	DORMAN

DICTIONARY FOR THE FIELD EXPERIENCE NEED CODES

CODE NUMBER	ITEM NAME
0	
1	ART EXPERIENCE
2	BETTER RAPPORT WITH STUDENTS
3	BETTER RAPPORT WITH STAFF
4	BETTER RAPPORT WITH PARENTS
5	BETTER UNDERSTANDING OF CHILDREN
6	BETTER VOICE CONTROL
7	CHALKBOARD EXPERIENCE
8	CONCERN FOR CHILDRENS WELFARE
9	DISCIPLINE EXPERIENCE
10	EVALUATION EXPERIENCE
11	EXPERIENCE DIRECTING LESSONS
12	EXPOSURE TO LOWER GRADES
13	EXPOSURE TO MORE GRADES
14	EXPOSURE TO UPPER GRADES
15	GROUP MANAGEMENT EXPERIENCE
16	LANGUAGE ARTS EXPERIENCE
17	LARGE GROUP EXPERIENCE
18	LOWER INCOME EXPERIENCE
19	MATH EXPERIENCE
20	MORE ENTHUSIASM
21	MORE PERSERVERENCE
22	MORE TEACHING TECHNIQUES
23	MUSIC EXPERIENCE
24	P.E. EXPERIENCE
25	POSITIVE ATTITUDE
26	READING EXPERIENCE
27	SCIENCE EXPERIENCE
28	SMALL GROUP EXPERIENCE
29	SOCIAL STUDIES EXPERIENCE
30	TEACHING EXPERIENCE
31	TESTING EXPERIENCE
32	TO ADAPT BETTER
33	TO BE AWARE OF INDIVIDUAL INTERESTS
34	TO BE AWARE OF INDIVIDUAL PROBLEMS
35	TO BE CONFIDENT
36	TO BE CREATIVE
37	TO BE DEPENDABLE
38	TO BE INNOVATIVE
39	TO BE LESS SENSITIVE
40	TO BE ORGANIZED
41	TO BE MORE SENSITIVE
42	TO BE PATIENT
43	TO BE PREPARED
44	TO BE PUNCTUAL
45	TO BE RESPONSIBLE
46	TO BE RELAXED
47	TO BE WILLING TO HELP CHILDREN
48	TO BE WILLING TO HELP COWORKERS
49	TO COMPLETE PROJECTS
50	TO CONSTRUCT EVALUATION INSTRUMENTS
51	TO CONTROL LARGE GROUPS
52	TO EMPHASIZE INDIVIDUAL NEEDS
53	TO HELP STUDENTS FEEL SUCCESSFUL
54	TO INITIATE CLASS INVOLVEMENT
55	TO PLAN BETTER
56	TO PREPARE UNITS
57	TO RECOGNIZE INDIVIDUAL NEEDS
58	TO RELATE BETTER
59	TO RESPOND BETTER TO CRITICISM
60	TO SHOW INITIATIVE
61	TO TEACH FROM CHILDRENS INTEREST
62	TO TREAT STUDENTS FAIRLY
63	TO USE OUTSIDE RESOURCES BETTER
64	TO USE PEER TEACHING
65	WORK IN CREATIVE AREAS

DICTIONARY FOR THE FIELD EXPERIENCE NEED CODES (CONTINUED)

CODE NUMBER	ITEM NAME
66	WORK WITH LEARNING PROBLEMS
67	WORK IN ALL AREAS
68	WORK WITH DIFFERENT MATERIALS
69	
70	

DICTIONARY FOR THE FIELD EXPERIENCE ROLE CODES

CODE NUMBER	ITEM NAME
0	
1	ADMINISTERED SOCIO-GRAM
2	ADMINISTERED STANDARDIZED TEST
3	ADMINISTERED TESTS
4	ARRANGED CLASSROOM ENVIORNMENT
5	ARRANGED SCHOOL VISIT FROM A RESOURCE PERSON
6	ATTENDED SCHOOL FUNCTION
7	DEVELOPED LEARNING CENTERS
8	DIRECTED PRODUCTION OF A PLAY OR PROGRAM
9	DIRECTED SHARING ACTIVITIES
10	ENRICHMENT ACTIVITIES
11	EVALUATED PAPERS
12	EVALUATED STUDENT PROGRESS
13	GEN. ACTIVITIES AND PROCEDURES
14	INDIVIDUAL INSTRUCTION
15	LARGE GROUP INSTRUCTION
16	LEAD CLASSROOM DISCUSSION
17	MADE HOME VISIT
18	MADE MATERIALS AND/OR DISPLAYS
19	OBSERVED CLASS
20	OBSERVED PSYCH SERVICES
21	OBSERVED TEAM TEACHING
22	PARENT TEACHER CONFERENCE
23	PLANNED ACTIVITIES AROUND UNITS
24	PLANNED AND SUPERVISED ACTIVITIES
25	PLANNED GAMES AND ACTIVITIES
26	PLANNED GROUP INSTRUCTION
27	PLANNED WORK TIME
28	PLAYED GAMES WITH CHILDREN
29	PREPARED INDEPENDENT WORK
30	PREPARED LESSONS FOR CLASS
31	READ STORIES TO CLASS
32	SMALL GROUP INSTRUCTION
33	SOME TEAM TEACHING
34	TAUGHT ART
35	TAUGHT LANGUAGE ARTS
36	TAUGHT MATH
37	TAUGHT MUSIC
38	TAUGHT P.E.
39	TAUGHT SCIENCE
40	TAUGHT SOCIAL STUDIES
41	TAUGHT READING
42	TEAM PLANNING
43	SMALL GROUP INSTRUCTION
44	USED AUDIO-VISUAL EQUIPMENT
45	WORKED IN ALL AREAS
46	WORKED WITH EMR CHILDREN
47	WORKED WITH FOREIGN STUDENTS
48	WORKED WITH GROUPS
49	WORKED WITH LEARNING CENTERS
50	USED DIFFERENT TEACHING TECHNIQUES
51	INDIVIDUAL CLASSROOM CONFERENCES
52	
53	
54	
55	

DICTIONARY FOR THE FIELD EXPERIENCE STRENGTH CODES

CODE_NUMBER	ITEM_NAME
0	
1	ADAPTS MATERIALS TO ACHIEVEMENT LEVELS
2	AWARE OF INDIVIDUAL INTERESTS
3	AWARE OF INDIVIDUAL NEEDS
4	AWARE OF INDIVIDUAL PROBLEMS
5	CAPABLE
6	CHILDREN LIKE HER(HIM)
7	CONCERNED WITH CHILDS WELFARE
8	CONFIDENT
9	CONSTRUCTS GOOD EVALUATION INSTRUMENTS
10	COOPERATIVE
11	CREATIVE
12	DEPENDABLE
13	DESIRE TO HELP CHILDREN
14	DESIRE TO LEARN
15	EFFECTIVE CLASSROOM CONTROL
16	EMPHASIZES INDIVIDUAL NEEDS
17	EVALUATES ACCURATELY
18	EXCELLENT PLANNER
19	EXCELLENT RAPPORT
20	EXCELLENT RELATION WITH STAFF
21	EXCELLENT RELATION WITH PARENTS
22	EXCEPTIONALLY CAPABLE IN ART
23	EXECPTIONALLY CAPABLE IN LANG.A.
24	EXCEPTIONALLY CAPABLE IN MATH
25	EXCEPTIONALLY CAPABLE IN MUSIC
26	EXCEPTIONALLY CAPABLE IN READING
27	EXCEPTIONALLY CAPABLE IN SCIENCE
28	EXECPTIONALLY CAPABLE IN S.S.
29	FLEXIBLE
30	GAINS STUDENTS TRUST
31	GOES BEYOND WHAT IS EXPECTED
32	GOOD RAPPORT
33	GOOD VOICE CONTROL
34	GOOD WITH CHILDREN
35	GROWTH IN DEPENDIBILITY
36	HELPS STUDENTS FEEL SUCCESSFUL
37	INNOVATIVE
38	INTEREST IN CHILDRENS WORK
39	INITIATES CLASS INVOLVEMENT
40	KNOWLEDGEABLE
41	LARGE GROUP WORK
42	LIKES THIS AGE GROUP
43	LIKES WORKING WITH CHILDREN
44	MOTIVATES AND INTERESTS STUDENTS
45	PATIENT
46	PERSERVERENCE
47	PLANS EFFECTIVELY
48	POSITIVE ATTITUDE
49	PROFESSIONAL PROMISE
50	PROMPT
51	RESPONDS WELL TO CRITICISM
52	SEES WHAT NEEDS TO BE DONE
53	SENSE OF RESPONSIBILITY
54	SENSITIVE TO STUDENTS FEELINGS
55	SHOWS INITIATIVE
56	SMALL GROUP WORK
57	SYMPATHETIC
58	TEACHES FROM CHILDRENS INTERESTS
59	TREATS CHILDREN FAIRLY
60	UNDERSTANDS CHILDREN
61	USES INDIVIDUAL DIFFERENCES
62	USES OUTSIDE RESOURCES WELL
63	USES PEER TEACHING
64	VERY RESPONSIBLE
65	WANTS CHILDREN TO UNDERSTAND CONCEPTS

DICTIONARY FOR THE FIELD EXPERIENCE STRENGTH CODES (CONTINUED)

CODE NUMBER	ITEM NAME
66	WELL ORGANIZED
67	WELL PREPARED
68	WORK WITH LEARNING PROBLEMS
69	WORK IN ALL PHASES OF CLASSROOM ACTIVITIES
70	EFFICIENT
71	ENTHUSIASTIC
72	FRIENDLY
73	GOOD COMMAND OF ENGLISH
74	GOOD IDEAS
75	POISED

DICTIONARY FOR THE GRADE LEVEL CODES

CODE NUMBER	ITEM NAME
1	FIRST
2	SECOND
3	THIRD
4	FOURTH
5	FIFTH
6	SIXTH
7	SEVENTH
8	EIGHTH
9	PRE-KINDERGARTEN
10	KINDERGARTEN
11	MULTI-AGE PRIMARY
12	MULTI-AGE UPPER EL
13	MIDDLE SCHOOL
14	SPECIAL EDUCATION
15	MUSIC
16	READING
17	PHYSICAL EDUCATION

DICTIONARY FOR THE GRADE POINT AVERAGE CODES

CODE NUMBER	ITEM NAME
0	
1	1.5 OR LESS
2	1.6 - 2.0
3	2.1 - 2.5
4	2.6 - 3.0
5	3.1 - 3.5
6	3.6 - 4.0

DICTIONARY FOR THE LEARNING ACTIVITY COMMENT CODES

CODE NUMBER	ITEM NAME
0	
1	ACCURATE
2	ADAPTABLE
3	BRIEF
4	CAREFUL IDEAS
5	COMMAND OF SUBJECT
6	COMPETENT
7	COMPLETE
8	CREATIVE
9	CRITICAL INDEPENDENT THINKER
10	DESCRIPTIVE
11	ENERGETIC
12	ENTHUSIASTIC
13	EXCELLENT
14	EXCELLENT IDEAS
15	FAIR
16	FLEXIBLE
17	GOOD
18	GOOD CLASSROOM APPLICATIONS
19	GOOD CRITERIA
20	IMAGINATIVE
21	INTERESTING
22	LOGICAL
23	MOTIVATED
24	NEED MORE EVIDENCE TO SUPPORT CONCLUSION
25	NEEDS MORE DOCUMENTATION
26	NEEDS TO BE MORE CONCISE
27	NOT ENOUGH SOURCES
28	ORIGINAL IDEAS
29	PRECISE
30	POOR
31	REASONED INFERENCE
32	RELIABLE
33	RELATES WELL TO PEOPLE
34	RESOURCEFUL
35	SATISFACTORY
36	SEQUITER
37	SCIENTIFIC
38	SCHOLARLY
39	STEADY PURSUIT OF TASK
40	STIMULATING
41	SYSTEMATIC
42	THOUGHTFULLY APPLIED
43	TOO GENERAL
44	WELL CHOSEN READINGS
45	WELL DISCUSSED
46	WELL DONE
47	WELL RESEARCHED
48	WELL STATED
49	WORK COMPETENT IF ATTENTIVE
50	WORK SHOWS HUMOR
51	WRITING DIFFICULT TO READ
52	
53	ABILITY TO DO THIS
54	INTEREST IN WORKING WITH PUPILS
55	NO PROBLEMS
56	UNDERSTOOD ACTIVITY
57	PASSED
58	WORKS HARD
59	INTERESTED
60	

DICTIONARY FOR THE LEARNING ACTIVITY TITLE CODES

CODE NUMBER	ITEM NAME
0	
1	ORIENTATION
2	FINAL EVALUATION
3	NEED AND ROLE OF ART
4	ART GROWTH AND DEVELOPMENT
5	ART CURRICULUM PLANNING
6	STUDIO ACTIVITIES
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	BASES FOR CURRICULUM PAPER
32	MEDIA SEARCH
33	SOCIAL CLASS INTERVIEWS
34	STUDY OF ONE CHILD
35	REPORTING PUPIL PROGRESS
36	TEACHING LIFE PROCESSES
37	CREATIVE THINKING MODULE
38	ORGANIZING LEARNING EXPERIENCES
39	ROLE OF TEACHER
40	GROUPING FOR INSTRUCTION
41	PLANS FOR TEACHING-LEARNING SETTING
42	DEVELOPING INSTRUCTIONAL STRATEGIES
43	
44	
45	
46	
47	
48	
49	
50	
51	LISTENING AND SPEAKING
52	NATURE OF LANGUAGE
53	VISUAL LITERACY
54	HANDWRITING AND SPELLING
55	COMPOSITION AND GRAMMAR
56	CHILDRENS LANGUAGE
57	SHARING LEARNING CENTERS
58	CREATIVITY
59	
60	
61	
62	
63	
64	
65	

DICTIONARY FOR THE LEARNING ACTIVITY TITLE CODES (CONTINUED)

CODE NUMBER	ITEM NAME
66	
67	
68	
69	
70	
71	BEGINNING NUMBERS
72	ADDITION - SUBTRACTION
73	MULTIPLICATION - DIVISION
74	FRACTIONS
75	DECIMALS AND PERCENT
76	MEASUREMENT AND GEOMETRY
77	NUMERATION SYSTEMS AND BASES
78	
79	
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
90	
91	ANCILLARY INSTRUMENTS
92	TEACHING
93	RESOURCE COLLECTION
94	INTERRELATED TEACHING UNIT
95	CARL ORFF APPROACH
96	CREATIVE COMPOSITIONAL APPROACH
97	ZOLTON KODOLY APPROACH
98	FUNDAMENTAL, CREATIVE AND DIRECTED MOVEMENT
99	LEARNING CENTERS AND IPBI IN MUSIC
100	PHILOSOPHY PAPER
101	MAKING AND PLAYING THE DULCIMER
102	COMPOSITIONAL APPROACH
103	MUSICAL CONCEPTS FOR EARLY CHILDHOOD
104	MODELS FOR TEACHING
105	MUSIC AND OPEN EDUCATION
106	ROCK,JAZZ, GOSPEL:' LET THEM TEACH
107	ORFF GUIDED PROJECT
108	
109	
110	ACTIVITY 1
111	ACTIVITY 2
112	ACTIVITY 3
113	ACTIVITY 4
114	ACTIVITY 5
115	ACTIVITY 6
116	ACTIVITY 7
117	ACTIVITY 8
118	ACTIVITY 9
119	ACTIVITY 10
120	APPLICATION OF PRINCIPLES OF HUMAN DEVELOPMENT
121	COMPLETED BY TAKING EDF345
122	
123	
124	
125	
126	
127	
128	
129	
130	TERMINOLOGY AND WORD ATTACK SKILLS
131	ASSESS PUPILS READING LEVELS

DICTIONARY FOR THE LEARNING ACTIVITY TITLE CODES (CONTINUED)

CODE NUMBER	ITEM NAME
132	SELECTION OF APPROPRIATE MATERIALS
133	DETERMINE APPROPRIATE SKILLS AND TEACH THEM
134	CLASSROOM ORGANIZATION AND RECORD KEEPING
135	PREPARATION OF MATERIAL FILE
136	READING READINESS
137	METHODS OF TEACHING READING
138	
139	
140	
141	
142	
143	
144	
145	
146	
147	
148	
149	
150	LABORATORY 1
151	LABORATORY 2
152	LABORATORY 3
153	LABORATORY 4
154	GRAPHING VARIABLES
155	MEASURING
156	DESIGNING AND RUNNING EXPERIMENTS
157	ECOLOGICAL VARIABLES
158	CONTROLLING VARIABLES
159	WAIT TIME AS A FACTOR IN INQUIRY
160	FATE CONTROL
161	VERBAL SANCTIONS
162	INQUIRY
163	CLASSROOM CONTROL
164	MANAGEMENT
165	EXEMPT BY PREVIOUS REQUIREMENTS
166	
167	
168	
169	
170	KEEPING UP WITH RESEARCH
171	SYSTEMATIC CLASSROOM OBSERVATION
172	MAPPING AND AREA SOCIAL SYSTEMS
173	COMPLETED BY TAKING EDF320
174	
175	
176	
177	
178	
179	
180	SOCIAL SCIENCE DISCIPLINES
181	MEDIA AND ACTIVITIES
182	SKILLS OF SOCIAL STUDIES
183	PROGRAM PLANNING AND EVALUATION
184	PERSONAL-SOCIAL AND INTERGROUP PROBLEM SOLVING
185	GROUP STUDIES AND PROBLEM SOLVING
186	
187	
188	
189	
190	ART
191	CURRICULUM
192	HUMAN GROWTH AND DEVELOPMENT
193	LANGUAGE ARTS
194	MATHEMATICS
195	MUSIC
196	READING
197	SCIENCE
198	SOCIAL FOUNDATIONS
199	SOCIAL STUDIES
200	

DICTIONARY FOR THE LEARNING ACTIVITY GROUP CODES

CODE NUMBER	ITEM NAME
1	GROUP
2	INDIVIDUAL

DICTIONARY FOR THE LEARNING ACTIVITY SUBSTITUTE CODES

CODE NUMBER	ITEM NAME
1	SUBSTITUTE
2	REGULAR

DICTIONARY FOR THE LEVEL IN CEP CODES

CODE NUMBER	ITEM NAME
0	
1	INITIATE
2	ASSISTANT
3	ASSOCIATE
4	INTENSIVE

DICTIONARY FOR THE MARITAL STATUS CODES

CODE NUMBER	ITEM NAME
0	
1	SINGLE
2	MARRIED
3	DIVORCED
4	WIDOWED

DICTIONARY FOR THE MYERS BRIGGS TYPE CODES

CODE NUMBER	ITEM NAME
0	
1	ISTJ
2	ISTP
3	ESTP
4	ESTJ
5	ISFJ
6	ISFP
7	ESFP
8	ESFJ
9	INFJ
10	INFP
11	ENFP
12	ENFJ
13	INTJ
14	INTP
15	ENTP
16	ENTJ

DICTIONARY FOR THE PRINT REQUEST CODES

CODE NUMBER	ITEM NAME
1	MYERS BRIGGS
2	REQ. COURSES
3	REGISTRATION
4	L.A.
5	L.A. SUMMARY
6	FIELD
7	FIELD SUMMARY
8	COMMUNITY

DICTIONARY FOR THE QUARTER NUMBER CODES

CODE NUMBER	ITEM NAME
0	
1	FALL
2	WINTER
3	SPRING
4	SUMMER

DICTIONARY FOR THE REQUIRED COURSE NAME CODES

CODE NUMBER	ITEM NAME
1	EH491
2	MSC260
3	MSC390
4	MS310/311
5	PHR361
6	PHR373
7	SCH201
8	MEDIA EQUIPMENT
9	MEDIA PRODUCTIONS

DICTIONARY FOR THE SCHOOL NAME CODES

CODE NUMBER	ITEM NAME
1	J.J. FINLEY
2	HAWTHORNE
3	SIDNEY LANIER
4	LITTLEWOOD
5	SHELL
6	TEWILLIGER
7	WESTWOOD

DICTIONARY FOR THE SEX CODES

CODE LETTER	ITEM NAME
F	FEMALE
M	MALE

DICTIONARY FOR THE SPECIALIZATION TYPE CODES

CODE NUMBER	ITEM NAME
1	ELEMENTARY
2	EARLY CHILDHOOD

```
DICTIONARY FOR THE TRAVEL EXPERIENCE CODES

CODE_NUMBER    ITEM_NAME
     0
     1         AFRICA
     2         ASIA
     3         BERMUDA AND BAHAMAS
     4         CANADA
     5         CARIBBEAN
     6         CENTRAL AMERICA
     7         EUROPE
     8         MEXICO
     9         MIDDLE EAST
    10         PACIFIC ISLANDS
    11         U.S.A.- ALL OVER
    12         FAR WESTERN STATES
    13         MIDDLE ATLANTIC STATES
    14         MIDWESTERN STATES
    15         NEW ENGLAND STATES
    16         PLAINS STATES
    17         ROCKY MOUNTAIN STATES
    18         SOUTHEASTERN STATES
    19         SOUTHWESTERN STATES
    20
```

```
DICTIONARY FOR THE TYPE CODES

CODE_NUMBER    ITEM_NAME
     0         NO OLD MAIN FILE EXISTS
     1         OLD MAIN FILE EXISTS
```

```
DICTIONARY FOR THE WORK EXPERIENCE CODES

CODE_NUMBER    ITEM_NAME
     0
     1         BABYSITTER
     2         CAMP COUNSELOR
     3         CLERK
     4         CLUB WORK
     5         DAY CARE CENTER
     6         LABORATORY TECHNICIAN
     7         LITTLE THEATRE
     8         NURSERY SCHOOL
     9         RECEPTIONIST
    10         RECREATION CENTER
    11         SALESPERSON
    12         SECRETARY
    13         STUDENT ASSISTANT
    14         SUBSTITUTE
    15         TEACHERS AID
    16         TOUR GUIDE
    17         TUTOR
    18         OTHER WORK CLASSIFICATION
```

BIBLIOGRAPHY

Baker, Frank B. "Computer-Based Instructional Management Systems: A First Look," Review of Educational Research, XLI (February, 1971), 51-70.

Blume, Robert. "Humanizing Teacher Education," Phi Delta Kappan, LII (March, 1971), 411-415.

Bonar, John R. and Dick, Walter. "Development of a Computerized Management Information System for Teacher Education Programs," Educational Technology, XIV (February, 1974), 41-44.

Busby, Walter A., Avila, Donald, Blume, Robert, Combs, Arthur W., and Oberlin, Lynn. "An Experiment in Teacher Education Utilizing the 'Self as Instrument' Concept," Gainesville, Florida: University of Florida, 1972 (mimeographed).

CEP. "CEP Student Handbook," Gainesville, Florida: Childhood Education Program, College of Education, University of Florida, 1973 (mimeographed).

Coffin, Robert W. "Appendix X -- Comfield Information Management System," in A Competency Based, Field Centered, Systems Approach to Elementary Teacher Education, Volume II -- Appendices A-L, Eds. Schalock, H. Del and Hale, James R. Washington: U.S. Government Printing Office, 1968.

Combs, Arthur W. The Professional Education of Teachers: A Perceptual View of Teacher Preparation. Boston: Allyn and Bacon, Inc., 1965.

Combs, Arthur W. and Kinzer, Suzanne M. "The University of Florida's Childhood Education Program: An Alternative Approach to CBTE," in Exploring Competency Based Education, Ed. Houston, W. Robert. Berkeley, California: McCutchan Publishing Corporation, 1974.

Combs, Arthur W., Soper, Daniel W., Gooding, C. Thomas et al. Florida Studies in the Helping Professions. University of Florida Social Science Monograph No. 37, Gainesville, Florida: University of Florida Press, 1969.

Cooley, William W. and Glaser, Robert. "An Information and Management System for Individually Prescribed Instruction," Working Paper 44, Pittsburgh: Learning Research and Development Center, 1968.

Data Element Dictionary for Student Information. Gainesville, Florida: Florida Community Junior College Inter-institutional Research Council, Institute of Higher Education, University of Florida, 1970.

Fiddleman, Richard H. and Gorman, Michael M. "PMIS: A Management Information System for Local School Districts," AEDS Journal, V (March, 1972), 80-90.

Flanagan, John C. "The Role of the Computer in PLAN," Journal of Educational Data Processing, VII (February, 1970), 7-17.

Geddes, Cleone L. and Kooi, Beverly Y. "An Instructional Management System for Classroom Teachers," Elementary School Journal, LXIX (April, 1969), 337-345.

Gentry, Castelle G. and Jurs, Stephen. "CBTE Management Information Manual," Toledo, Ohio: University of Toledo (mimeographed, n.d.).

Goodlad, John I., O'Toole, John F., Jr., and Tyler, Louise L. Computers and Information Systems in Education. New York: Harcourt, Brace & World, Inc., 1966.

Hamreus, Dale J. "Specifications for a System to Manage a ComField Based Instructional Program," in A Competency Based, Field Centered, Systems Approach to Elementary Teacher Education, Volume I, Eds. Schalock, H. Del and Hale, James R. Washington: U.S. Government Printing Office, 1968.

Hansen, Duncan H. and Harvey, William L. "Impact of CAI on Classroom Teachers," Educational Technology, X (February, 1970), 46-48.

Hayman, John L., Jr. "Educational Management Information Systems for the Seventies," Educational Administration Quarterly, X (Winter, 1974), 60-71.

Hayman, John L., Jr. and Mable, Cynthia. "Computer Managed Instruction at a College of Education," Educational Technology, XIV (September, 1974), 60-63.

Healy, John. "COMPUTRAC: Monitoring Student Classroom Performance," AEDS Journal, IV (March, 1971), 69-74.

Holzman, G. and Turksen, I.B. "Computerizing Student Information in an Engineering School," Engineering Education, LXI (January, 1971), 377-378.

Kennevan, Walter J. "MIS Universe," Data Management, VIII (September, 1970), 62-64.

Lawrence, Ben. "The Western Interstate Commission for Higher Education Management Information Systems Program," in Management Information Systems in Higher Education: The State of the Art, Eds. Johnson, Charles B. and Katzenmeyer, William G. Durham, North Carolina: Duke University Press, 1969.

Lawrence, Ben and Gulko, Warren W. "A National Effort to Improve Higher Education Management," Journal of Educational Data Processing, IX, 1-2 (1972), 44-54.

Lawrence, G. Benjamin, Lovell, Clifton C., and Lyell, Edward H. "The Implications for Educational Research of Emerging Planning and Management Systems in Higher Education," Journal of Educational Research, LXIV (March, 1971), 291-296.

Levin, Howard S. Office Work and Automation. New York: John Wiley & Sons, Inc., 1956.

Lorber, Michael A. "Increased Learning Freedom Via Computers," Paper presented at the Conference on Computers in the Undergraduate Curricula, Claremont, California (June 18-20, 1973), ERIC ED 081229.

Mathews, Walter M., Christopher, David F., and Bishop, Billy M. "Management Information Systems: The State of the Art," Paper presented at the American Institute for Decision Sciences Annual Meeting, New Orleans, La. (November 1-4, 1972), ERIC ED 090692.

Molnar, Andrew R. Computer Innovations in Education. Washington, D.C.: National Science Foundation, 1972.

Neuhauser, Charlotte. "The Design and Implementation of a Management Information System to Facilitate the Functioning of a CBTE Program," Paper presented at the Annual Meeting of the American Educational Research Association, Chicago (April, 1974).

Neuhauser, Charlotte. "A Management Information System to Facilitate the Functioning of a CBTE Program," Educational Technology, XV (March, 1975), 36-37.

Rebhun, H. F., Bonar, J. R., and Dick, W. "A Computerized Advisement Support-System and Research Data Base for Colleges of Education," Journal of Educational Data Processing, XI, 3 (1974), 18-28.

Schalock, H. Del. "Overview," in A Competency Based, Field Centered, Systems Approach to Elementary Teacher Education, Volume I, Eds. Schalock, H. Del and Hale, James R. Washington, D.C.: U.S. Government Printing Office, 1968.

Silberman, Harry F. "Computer-Managed Instruction: Use of a Computer to Monitor Student Performance," Engineering Education, LIX (February, 1969), 484-486.

Stefanu, Constantine, Fedinec, Alexander A., Farmer, T. Albert, and Pate, Mary L. "A Computerized Student Evaluation Information Retrieval System," Journal of Medical Education, L (February, 1975), 201-203.

Teichroew, Daniel and Sayani, Hasan. "Automation of System Building," Datamation, XVII (August 15, 1971), 25-30.

Usher, Richard and Hanke, John. "The 'Third Force' in Psychology and College Teacher Effectiveness Research at the University of Northern Colorado," Colorado Journal of Education Research, X (Winter, 1971), 2-9.

Van Dusseldorp, Ralph. "Some Principles for the Development of Management Information Systems," in Management Information Systems in Higher Education: The State of the Art, Eds. Johnson, Charles B. and Katzenmeyer, William G. Durham, North Carolina: Duke University Press, 1969.

Van Dusseldorp, Ralph. "Management Responsibility for Information Systems," Educational Technology, XI (May, 1971), 38-40.

Vinsonhaler, John F., Millin, Daniel, and Gillmore, Gerald M. "Computer Management of Instructional Resources," Educational Technology, XI (May, 1971), 32-35.

Vitulli, Ramon A. and Singleton, Roger L. "Computer Assisted Advising and Degree Evaluation," College and University, XLVII (Summer, 1972), 494-502.

Wass, Hannelore, Blume, Robert A., Combs, Arthur W., and Hedges, William D. Humanistic Teacher Education: An Experiment in Systematic Curriculum Innovation. Fort Collins, Colorado: Shields Publishing Co., Inc., 1974.

Webb, Rodman B. and Guinagh, Barry J. "The Translation of Theory into Practice: A Study of Student's Perceptions of the Childhood Education Program," Gainesville, Florida: University of Florida, 1975 (mimeographed).

BIOGRAPHICAL SKETCH

Bernice Doerr was born September 24, 1946, at Montclair, New Jersey. She attended schools in Newark and Springfield, New Jersey and was graduated from Jonathan Dayton Regional High School in Springfield in June, 1964.

After completing one year at Baldwin Wallace College in Berea, Ohio, she entered the University of Florida. In March, 1968, she received the Bachelor of Science in Education with a major in mathematics. At this time she was elected to the society of Phi Kappa Phi.

Miss Doerr continued her education at the University of Florida and received the Master of Education in June, 1969. During this period, she worked as a graduate assistant for the Research Service Center in the College of Education where she did computer programming and data processing of test data for local school systems.

For the next year, Miss Doerr worked as a programmer for Bell Telephone Laboratories in Holmdel, New Jersey. In September, 1970, she returned to the University of Florida to enter doctoral studies in Foundations of Education. She accepted a graduate assistantship with the Research Service Center. Later, in addition to that part-time position, she worked for the Intercollege Department of Computer and Information Sciences where she taught programming to both beginning and advanced students. Presently Miss Doerr is employed as a research associate for Project Follow Through.

Miss Doerr is a member of the American Educational Research

Association, the Association for Computing Machinery, the Association for Education Data Systems, and Kappa Delta Pi.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Wm. D. Hedges
William D. Hedges, Chairman
Professor of Education

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Vynce A. Hines
Vynce A. Hines
Professor of Education

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Frank D. Vickers
Frank D. Vickers
Associate Professor of Engineering Technology

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

William B. Ware
William B. Ware
Associate Professor of Education

This dissertation was submitted to the Graduate Faculty of the College of Education and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August, 1975

B. L. Sharp by MC Beller

Dean, College of Education

Dean, Graduate School